

KFKI-1980-44

SZÖKE J.  
LÁNG E.

ADATFELDOLGOZÁS

17.

TELJESEN TORZÍTOTT EGYKOMPONENSŰ  
EXPONENCIÁLIS LECSENGÉSI GÖRBÉK ANALÍZISE  
ITERÁCIÓS ELJÁRÁSSAL

*Hungarian Academy of Sciences*

CENTRAL  
RESEARCH  
INSTITUTE FOR  
PHYSICS

BUDAPEST



## PAPERS PUBLISHED IN THIS SERIES:

- 1/ J. Szőke, L. Varga, I. Nagypál: Experimental and Computer Analysis of Spectral Fine structure  
XIV. Coll. Spectr. Internatl. Debrecen 1967  
Proc. Conf. 1205 old.
- 2/ E. Dudar, J. Szőke: Generation of UV Spectra  
Proc. of Coll. Spectr. Internatl. Heidelberg 1971, 298 old.
- 3/ J. Szőke: Digital Spectroscopic Laboratory and Computerized Spectrum Library  
Proc. XVI. Coll. Spectr. Internatl. Heidelberg 1971  
A. Hilger 1971, 321 old.
- 4/ Szőke J.: Komputeres mérés és adatfeldolgozás a kémiában  
KFKI Report 1972-6
- 5/ Szőke J.: Komputeres adatfeldolgozás  
Magyar Kémikusok Lapja 1972, 27, 67
- 6/ J. Szőke: On-line Measurements and Computerized Data Processing of Spectra  
KFKI Report 72-5
- 7/ J. Szőke: Computer Analysis of Spectra by Deconvolution  
Chem. Phys. Lett. 1972, 15, 404; KFKI-Report 1972-29
- 8/ Szőke J.: Program for elimination of instrument distortions and improvement of resolution  
KFKI Report 1972-74
- 9/ J. Szőke, I. Horváth, I. Szilágyi: Determination of the Genuine Spectrum Measured by Grating Spectrometer  
Proc. of Coll. Spectr Internatl. XVII Firenze 1973 440 old.
- 10/ Follmann P., Heszberger I., Faludi A., Szőke J.: Az oftalmodiogram számítógépes értékelése  
Szemészet 1973, 110, 283
- 11/ Szőke J.: Radiativ lecsengési görbék értékelése  
KFKI-Report 1978-66  
Lumineszcencia kutatások aktuális kérdései /Szerk. Kozma L., Molnár M., Várkonyi Z. /Szeged 1978/

ADATFELDOLGOZÁS

17.

TELJESEN TORZÍTOTT EGYKOMPONENSŰ  
EXPONENCIÁLIS LECSENGÉSI GÖRBÉK ANALÍZISE  
ITERÁCIÓS ELJÁRÁSSAL

Szőke J., Láng E.

Központi Fizikai Kutató Intézet  
1525 Budapest 114, Pf. 49

HU ISSN 0368-5330  
HU ISSN 0209-6099  
ISBN 963 371 675 6



## KIVONAT

A készülékfüggvényével konvolúciósan torzított, egykomponensű, exponenciális lecsengési görbék fittelését tanulmányoztuk háromféle iterációs módszerrel.

A becsléspontosságu, kiindulási lecsengési időt momentum módszerrel határozzuk meg.

Az egyoldali iterációnál (DEC-IT-ONE) - a lecsengési időhöz közelítve - fokozatosan csökkentjük a  $\tau$  értéket, az eltérés négyzetösszeg kontrollja mellett.

A kétoldali iterációs eljárásnál (DEC-IT-TWO) három  $\tau$  és eltérés négyzetösszeg adatpárt használunk, s a következő próba- $\tau$  értékét az eltérés négyzetösszeggel súlyozott átlagképzés módszerével határozzuk meg. A legkisebb eltérés négyzetösszegű  $\tau$ -t mindig a középső elem tartalmazza.

A DEC-IT-TWO három adatpárjának másodfoku fittelésével dolgozó DEC-IT-2ND módszerrel a másik két módszernél csak egy nagyságrenddel durvább közelítés érhető el.

A DEC-IT-ONE és DEC-IT-TWO módszerek segítségével a  $\tau$  érték, kis szóhosszuságu számítógépen, általában 10/00 pontosan meghatározható. A DEC-IT-ONE esetében 15-20, a DEC-IT-TWO esetében 8-12 iterációs lépés szükséges a legjobb fittelés eléréséhez.

A közleményben ismertetjük speciális, nagy pontosságu, integrálási eljárásunkat (SIMPSON-SPECIAL). Mivel ez rendkívül időigényes, a konvolúciós integrál kiszámítására a GRINWALD-STEINBERG rekurziós formulát tartjuk ideálisnak.

## АННОТАЦИЯ

Тремя различными итерационными методами нами изучалась согласованность однокомпонентной кривой с экспоненциальным затуханием конволюционно искаженной функцией прибора.

Приближенное исходное значение  $\tau$  времени затухания было определено методом момента.

При односторонней итерации /DEC-IT-ONE/ - приближаясь к значению времени затухания - постепенно уменьшали значение  $\tau$  с одновременным контролем суммы квадрата расхождения.

При применении метода двусторонней итерации /DEC-IT-TWO/ использовали три пары данных значений  $\tau$  и суммы квадрата расхождений.

Следующее пробное значение  $\tau$  определялось методом образования средней величины, взвешенной суммой квадратов расхождений. Значение  $\tau$  с наименьшей суммой квадратов расхождений всегда содержится в среднем элементе.

Методом DEC-IT-2ND, использующим подгонку второй степени трех пар данных DEC-IT-TWO, можно достичь результат на порядок более грубый, чем двумя другими методами.

С помощью методов DEC-IT-ONE и DEC-IT-TWO на ЭВМ с малой длиной слов значение  $\tau$  обычно может быть определено с точностью 10/00. Для достижения наилучшей подгонки в случае метода DEC-IT-ONE требуется 15-20 шагов, а в случае метода DEC-IT-TWO - 8-12.

В статье описывается разработанный нами специальный высокоточный метод интегрирования /SIMPSON-SPECIAL/. Для расчета конволюционного интеграла, требующего очень много времени, идеальной мы считаем рекуррентную формулу GRINWALD-STEINBERG.

## ABSTRACT

Single exponential fitting of pulse excited decay curves distorted by an instrument function has been investigated by three different iterational methods.

The starting value of  $\tau$  (decay time) is assessed by moment-method.

At the one-sided iteration (DEC-IT-ONE) try- $\tau$  is successively decreased under RMS control, coming near to the real decay time. The two-sided iteration (DEC-IT-TWO) uses three pairs of  $\tau$ -s and RMS-s. The next try- $\tau$  is determined by the method of RMS weighted average. The least RMS ( $\tau$  pair) is always the middle value.

DEC-IT-2ND uses also three pairs of  $\tau$ -s and RMS-s to the second order polynomial fitting. The precision of the decay time determination with DEC-IT-ONE and DEC-IT-TWO is about 0.1 per cent and with DEC-IT-2ND is about 1 per cent. The required iteration numbers are: DEC-IT-ONE 15-20; DEC-IT-TWO 8-12; DEC-IT-2ND 5-8.

For the calculation of the convolution integral the GRINWALD-STEINBERG approximating recursion formula is recommended because the special precise integration procedure (SIMPSON-SPECIAL) described here is very time consuming.



T A R T A L O M

1. Bevezetés	1
2. A torzított lecsengési görbék matematikai leírása	3
3. A lecsengési görbék alakjának változása $\tau$ függvényében	3
4. A lecsengési idő becslése	5
5. Iterációs eljárások	6
5.1 Az egyoldali közelítés	6
5.2 A kétoldali közelítés	10
5.21 A másodfoku közelítés módszere	13
5.22 A súlyozott középértékek módszere	13
5.3 Az iterációs algoritmusok értékelése	15
6. A normálási faktor (amplitudófaktor) meghatározása	15
7. A konvolúciós integrál számítása	20
7.1 A konvolúciós integrál meghatározása numerikus módszerekkel	20
7.2 A konvolúciós integrál meghatározása rekurziós formulával	22
8. A program	23
9. A DEC-IT- programlista	26
10. A DEC-IT- programok használata	28
11. Futtatási mintapéldák	33
11.1 DEC-IT-ONE futtatási mintapélda	33
11.2 DEC-IT-TWO futtatási mintapélda	43
12. Köszönetnyilvánítás	47
13. Irodalom	47







## 1. BEVEZETÉS

Az egykomponensű, exponenciális karakterű, lecsengési görbék analízisére tökéletes megoldást biztosít a lineáris legkisebb négyzetek módszere, ha a logaritmált függvénynek van olyan szakasza, amelyen a készülékfüggvény (pl. gerjesztő pulzus okozta) torzítások elhanyagolhatók és a jel/zajviszony kielégítő [1]. Ez az eljárás teljesen torzított lecsengési görbéknél nem használható. Az ilyen feladatok iterációs eljárással oldhatók meg egyértelműen, ha feltételezzük, hogy a lecsengési görbe egykomponensű.

Annak ellenére, hogy számos olyan eljárás ismeretes, amelyekkel többkomponensű lecsengési görbék eredményesen analizálhatók, s ezek az eljárások alkalmasak az egykomponensű görbék kiértékelésére is, mégis célszerű az iterációs eljárást alkalmazni, mivel (i) egyik komplexebb módszer sem tekinthető abszolút eljárásnak, (ii) az iterációs eljárással - ha a feltételek adottak - a kiértékelés tetszőleges pontossággal elvégezhető, (iii) a módszer könnyen kézben tartható, és áttekinthető, (iv) fontos annak ismerete, hogy a lecsengési görbét egykomponensű közelítésben milyen jól tudjuk leírni. Az utóbbi különösen fontos többkomponensű exponenciális vagy nem-exponenciális lecsengési görbék analízisének, amikor a rosszul fittelt tartományok segítségével következtethetünk a lecsengési folyamat természetére.

Az iterációs eljárást már több kutató alkalmazta a lecsengési görbék kiértékelésére [2,3]. A jelen munka egy része annak a közleménysorozatnak, amelyek az impulzusüzemű mérési eljárásokkal nyert lecsengési görbék kiértékelési módszereinek standardizálására irányulnak.

A program TPA-i kissetítőgépen FOKAL nyelven készült.







## 2. A TORZÍTOTT LECSENGÉSI GÖRBÉK MATEMATIKAI LEÍRÁSA

Ha a  $K(j)$  gerjesztő impulzus-függvény félérték-szélessége összemérhető a  $\tau$  lecsengési idővel, a közelítés alapegyenlete

$$(1) \quad C(t) = A \int_{j=0}^t K(j) * \exp [-(t-j) * H/\tau] dj$$

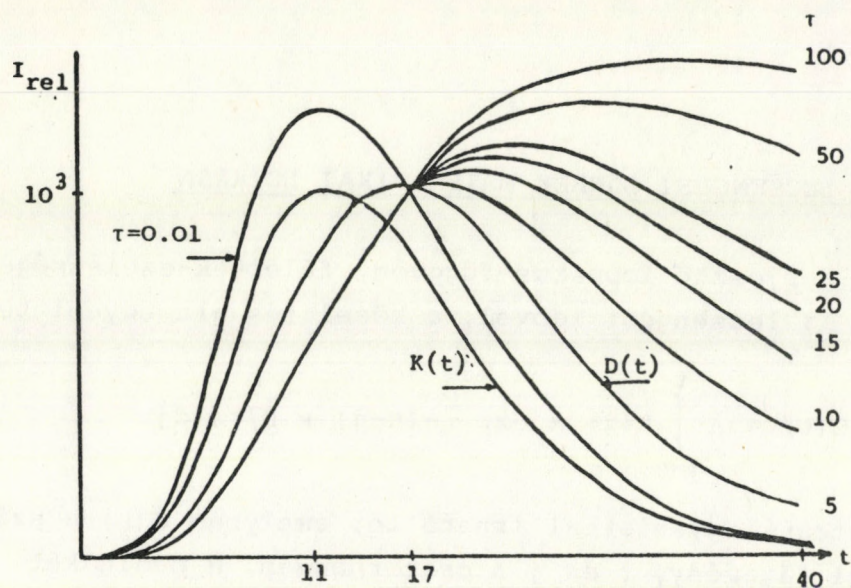
konvolúciós integrálegyenlettel írható le, amelyben  $C(t)$  a számított lecsengési függvény,  $t$  és  $j$  a csatornaszám,  $H$  pedig két szomszédos csatorna időkülönbsége (csatornaszélesség). Az (1) integrálegyenletet a 7. pontban közölt eljárások valamelyikével oldhatjuk meg.

## 3. A LECSENGÉSI GÖRBÉK ALAKJÁNAK VÁLTOZÁSA $\tau$ FÜGGVÉNYÉBEN

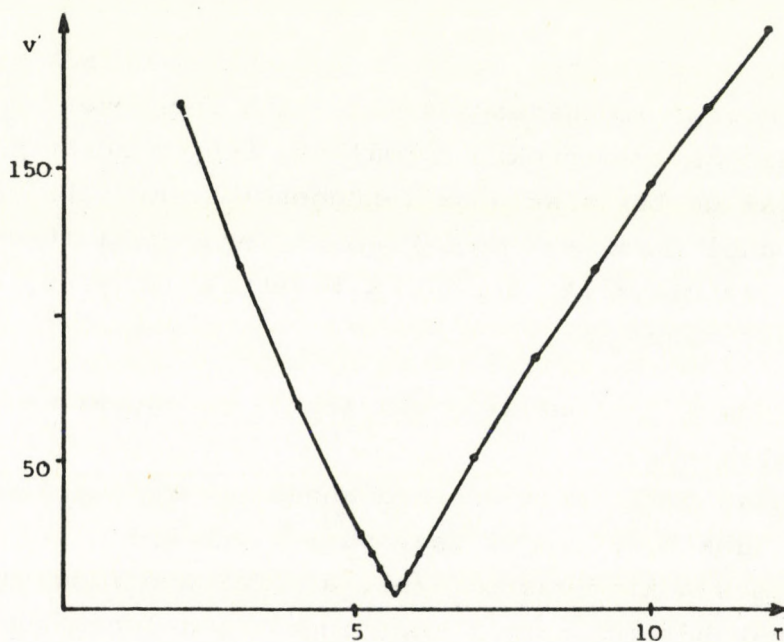
Adott gerjesztő függvénnyel keltett lecsengési görbék alakját csupán a lecsengési idő határozza meg. Ha a lecsengési görbék egy pontját azonos számértékre normáljuk, benyomásokat kaphatunk a lecsengési görbék alakjának  $\tau$ -függéséről. Ezt ábrázoltuk az 1. ábrán, ahol normálási pontnak a mért lecsengési függvény maximumhelyét választottuk. Az ábrából a következő általános következtetéseket vonhatjuk le:

- $\tau = 0$  esetén a lecsengési görbe alakja megegyezik a gerjesztő pulzusával;
- A lecsengési görbe maximumhelye annál inkább eltolódik a hosszabb idők felé, minél nagyobb a  $\tau$  értéke;
- Egy alkalmasan kiválasztott (pl. a gerjesztő függvény maximumának háromszoros)  $t$  koordinátánál a lecsengési görbék annál nagyobb intenzitásúak, minél nagyobb  $\tau$  értéke.





1. ábra. Az élettartam paraméter hatása a lecsengési görbe alakjára



2. ábra. A 2. egyenlettel definiált eltérésnégyzetösszeg változása az élettartam függvényében



d) Az 1. ábrán alkalmazott normálási feltételnél, ha a számított  $\tau$  kisebb, mint a kísérleti, akkor a  $C(t)$  görbén a normálási pont előtt a kísérleti görbe felett, utána pedig alatta halad. Ha a számított  $\tau$  nagyobb, mint a kísérleti, a helyzet fordított.

A  $N$  adatpárból álló  $D(t)$  kísérleti és  $C(t)$  számított görbék

$$(2) \quad v = \frac{1}{N} \sum_{t=0}^N [C(t) - D(t)]^2$$

átlagos eltérés-négyzetösszege a választott  $\tau$  értékek függvényében, két kvázi-lineáris függvénnyel jellemezhető (2. ábra), amelynek metszéspontja a zéró közelében van (a mérési zaj miatt nem zéró!) és az időkoordinátája a legjobb  $\tau$  érték. A minimumban a  $v$  a mérési zaj átlagértékével egyenlő.

Mindezek alapján biztatónak látszik, hogy lineáris paraméter javító algoritmussal eredményesen meg tudjuk oldani a  $\tau$  meghatározását [4].

#### 4. A LECSENGÉSI IDŐ BECSLÉSE

Az iterációs eljárás gyorsítása miatt célszerű egyszerű, becsléspontosságu lecsengési idő meghatározást is beépíteni a programba. E célra a legmegfelelőbbnek a momentum módszer látszik.

Brody [5] volt az első, aki a momentumok módszerét használta a lecsengési idő meghatározására. Bár sokan mások részletesen elemezték azt a módszert [6,7,8] és igyekeztek szélesebb körűvé tenni alkalmazhatóságát [9,10] véleményünk Demas-Crosbyval [3] egyezően a momentum módszer csak becsléspontosságu eredmények szolgáltatására alkalmas, amely után más, pl. egykomponensű rendszernél az iterációs módszer adja meg a legjobban fittelő eredményt.

A jelen esetben a momentum módszer legegyszerűbb formáját használjuk [3]

$$(3) \quad \tau_b = \frac{MD(1)}{MD(0)} - \frac{MK(1)}{MK(0)}$$



ahol MD a lecsengési görbe, MK a készülékfüggvény momentumát jelöli, amelyek indexei a momentumok rendjére utalnak. Egy  $F(t)$  függvény momentumait a következő integrálegyenlettel definiáljuk:

$$(4) \quad MF(I) = \int_0^{\infty} t^I F(t) dt$$

A diszkrét pontokból álló  $K(j)$  és  $D(j)$  függvények esetén numerikus integrálást végzünk (lásd 7. pont). Ha a 4. integrálegyenletet a legegyszerűbb módon, téglányszabállyal oldjuk meg, úgy a  $MK(I)$  és  $MD(I)$  momentumokat az (5) és (5a) egyenletek alapján nyerjük:

$$(5) \quad MK(I) = H \times \sum_{J=0}^N (J \times H)^I \times K(J)$$

$$(5a) \quad MD(I) = H \times \sum_{J=0}^N (J \times H)^I \times D(J)$$

A momentumszámításokat a program 10-es utasításcsoportja végzi.

## 5. ITERÁCIÓS ELJÁRÁSOK

A 2. ábra szerint az eltérés-négyzetösszeg minimális értékű a kísérleti  $\tau$ -koordinátánál. A feladat megoldására ezért eredményesen használhatjuk az iterációs eljárások valamelyikét. Elvileg kétféle közelítési mód áll rendelkezésre: (i) egyoldali és (ii) kétoldali közelítés módszere közül választhatunk. Mindkét közelítéstípusra készítettünk programot, s tapasztalatainkat az alábbiakban összegezzük:

### 5.1 Az egyoldali közelítés

Elvileg a valódi lecsengési idő mindkét oldalát felhasználhatjuk az iterációs eljáráshoz. A zéró-idő felőli oldalon biztos pont a zéró, de a lecsengési idő becsléspontosságu ismerete ebben az esetben is nélkülözhetetlen. A 2. ábra jól szemlélteti,



hogy ezen az oldalon az eltérésnégyzetfüggvény meredekebben változik, mint a valódi  $\tau$ -nál nagyobb idők oldalán.

Mindkét oldali közelítésnél az a lényeges, hogy egyre kisebb lépéseket alkalmazva addig haladunk, amíg a csökkenő eltérésnégyzetösszeg éppen növekedni kezd. Ekkor ugyanis átkerültünk a másik oldalra. Ezután kettőt visszalépve és csökkentve a lépésközt megismételjük az előbbi eljárást, amelyet addig folytatunk, amíg a valódi  $\tau$ -t átlépő lépéshossz nem válik elég kicsivé.

A vázolt eljáráshoz hasonló iterációs módszert alkalmaztak Hundley és munkatársai [2] is. Nagy értékről indulva a próba  $\tau$  értékét először nagy lépésekben (pl. nsec-enként) csökkentették, majd dekádikusan csökkentették a lépéshosszat. Ez az eljárás elvben megoldja a problémát, csak nagyon sok iterációs lépés szükséges a kívánt pontosság elérésére. Ezért célszerű volt megvizsgálni az iterációs ciklus-szám (= számolási idő) csökkentésének lehetőségét.

Mindkét oldali közelítésre készítettünk programot és lényegében a két közelítést egyenértékűnek találtuk. (Ez a megállapítás igaz a Hundley-féle közelítési módra is.) Ezért csak a hoszsú idők oldalán alkalmazott közelítést ismertetjük.

- a) Az iteráció megkezdése előtt a momentum módszerrel kiszámoljuk a becsléspontosságu  $\tau_0$  értékét. Ennek kétszeresét vesszük az első próba  $\tau_1$  értékének

$$(6) \quad \tau_1 = 2 \times \tau_0$$

- b) kiszámoljuk a DT lépésköz értékét, amelyet a  $\tau_1$  0,4-szeresének veszünk

$$(7) \quad DT = .4 \times \tau_1$$

- c) A  $\tau_1$ -gyel elvégezzük a fittelést. A próba  $\tau_1$  és  $v_1$  eltérés-négyzetösszeg értékét tároljuk az 5 elemes  $(-1, 0 \dots 3)$  TA(I) és SD(I) 3. elemeiben.

- d) A következő K-adik  $\tau$  értéket a

$$(8) \quad \tau_k = \tau_{k-1} - DT/K$$



összefüggéssel számoljuk, ahol  $K$  az iterációs számot jelent, majd elvégezzük a fittelést. Az előző próba  $\tau$  és eltérés-négyzetösszeg értékét 1-gyel kisebb sorszámú helyre léptetjük és a 3. elemeket felülírjuk a második iteráció eredményével.

- e) Az eljárást addig folytatjuk, amíg a  $K$ -edik eltérés-négyzetösszeg nem válik nagyobbá a  $(K-1)$ -ediknél. Ekkor
- f) - az iterációs lépéseket újra kezdjük számolni 1-től;  
- új  $DT$  értéket határozzunk meg

$$(9) \quad DT = |v_{K-2} - v_K| \times .4$$

- és a  $TA(I)$  ill.  $V(I)$  tömbökben tárolt adatokat 2-vel a nagyobb indexű elemekbe shifteljük.
- kiírja az adott közelítés eredményét a minimális eltérés-négyzetösszeghez tartozó  $\tau$  megadásával (ez a  $\tau_{K-1}$ -gyel egyenlő) a következő formában:

$$(10) \quad \text{DECAY TIME} = \tau_{K-1} (+/-) \sigma$$

ahol  $\sigma$  a megadott  $\tau$  pontossága, amelyet a következő egyenlettel definiálunk:

$$(11) \quad \sigma = (\tau_{K-2} - \tau_K) / 2$$

- a program interaktív kérdésére eldöntjük, hogy folytatjuk-e tovább az iterációs eljárást?  
Ha a folytatás mellett döntünk a program visszalép a c) ponthoz és halad az f) pont felé.

Az egyoldali közelítés futtatási mintapéldáját a 10. pontban tüntettük fel. Az eljárás blokkvázlata a 3. ábrán látható.

A DEC-IT-ONE Fokál nyelvű program 1-es utasítástömbjének speciális vonatkozásai:

C-FOKAL, 1971 KE

01.04 T ! "DEC-IT-TRI

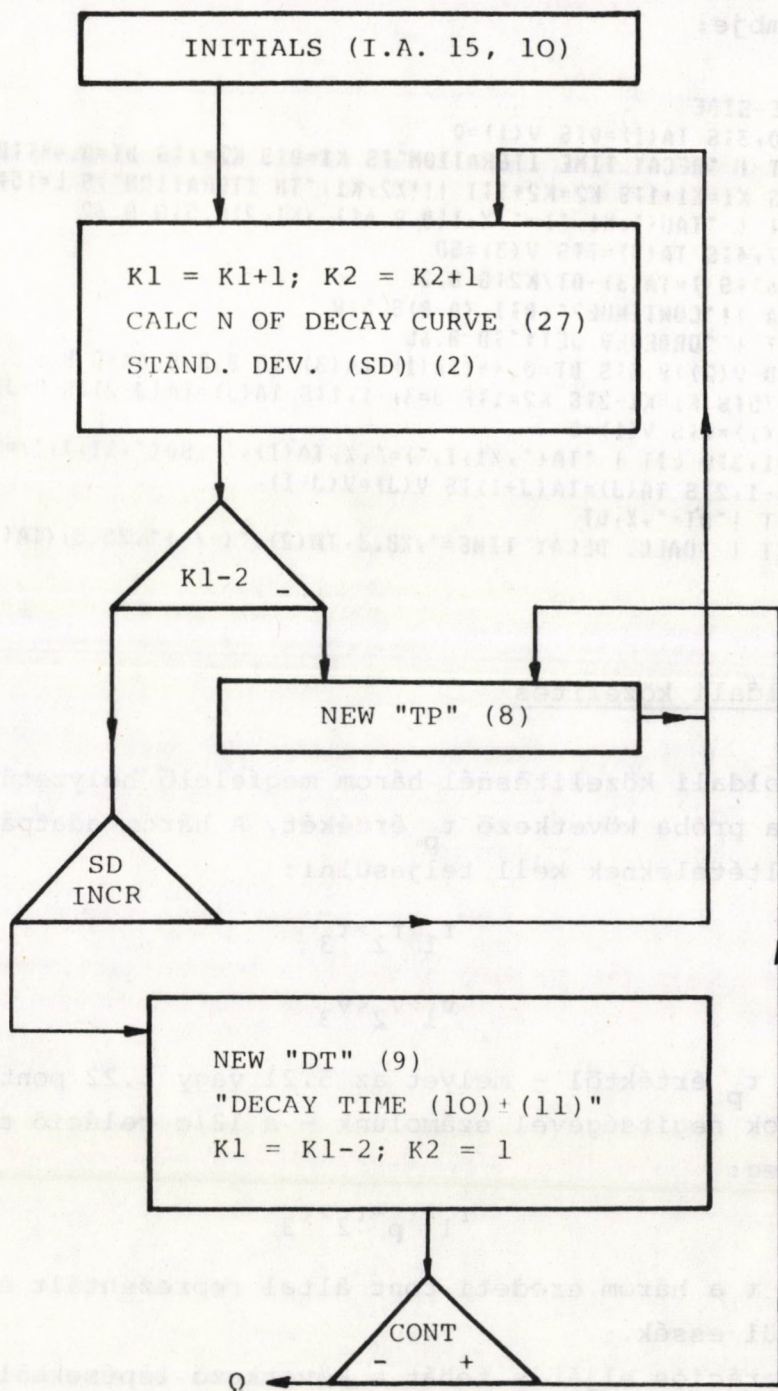
01.05 C "PROGRAM FOR CALCUL OF THE RADIATIVE LIFETIME "

01.06 C "OF THE EXCITED STATE USING TRIANGLE METHOD

01.07 C WRITTEN BY DR J. SZOKE - IN 1980

01.08 C CENTRAL RESEARCH INSTITUTE FOR PHYSICS - 1525 BUDAPEST POB 49





3. ábra. A DEC-IT-ONE eljárás funkcionális blokkvázlata

Jelölések: I.A. = Instruction Array /zárójelben a tömb sorszáma/,  
a( )-ben levő szám a közlemény megfelelő egyen-  
letének számát jelenti;

INCR = az előző ciklushoz képest növekvő



A DEC-IT-ONE Fokai nyelvű program egyoldali iterációs (8-as) utasítástömbje:

```

08.05 C ONE-SIDE
08.08 F 1=0,3;S TA(I)=0;S V(I)=0
08.10 O L;T ! "DECAY TIME ITERATION";S K1=0;S K2=1;S DT=0.4*T;D 8.75
08.20 O L;S K1=K1+1;S K2=K2+1;T !!!X2,K1,"TH ITERATION";S L=15;D 12.5
08.24 O L;T ! "TAU(",K1,")=",X,T;D 8.4;I (K1-2)8.5;G 8.62
08.40 D 8.7,4;S TA(3)=T;S V(3)=SD
08.50 D 8.61;S 1=TA(3)-DT/K2;G 8.2
08.60 O T;A !! "CONTINUE?";B;I (0-B)8.5;R
08.61 O L;T ! "ORDERED SET:";D 8.68
08.62 I (SD-V(2))8.5;S DT=0.4*(TA(1)-TA(3));D 8.8,8.64;G 8.6
08.64 D 8.75;S K1=K1-2;S K2=1;F J=3,-1,1;S 1A(J)=TA(J-2);S V(J)=V(J-2)
08.66 S 1A(1)=0;S V(1)=0
08.68 F 1=1,3;O L;T ! "TA(",X1,I,")=",X,TA(I)," SD(",X1,I,")=",X,V(I)
08.70 F J=-1,2;S TA(J)=TA(J+1);S V(J)=V(J+1)
08.75 O L;T ! "DT=",X,DT
08.80 O L;T ! "CALC. DECAY TIME=",X8.3,TA(2),"(+/-)",X5.3,(TA(1)-1A(3))/2

```

## 5.2 A kétoldali közelítés

A kétoldali közelítésnél három megfelelő helyzetű adatpárból számoljuk a próba következő  $\tau_p$  értékét. A három adatpárra a következő feltételeknek kell teljesülni:

$$(12a) \quad \tau_1 > \tau_2 > \tau_3$$

$$(12b) \quad v_1 > v_2 < v_3$$

Az új  $\tau_p$  értéktől - melyet az 5.21 vagy 5.22 pontokban leírt algoritmusok segítségével számolunk - a 12/c reláció teljesülését kívánjuk meg:

$$(12c) \quad \tau_1 > \tau_p \approx \tau_2 > \tau_3$$

azaz az új  $\tau$  a három eredeti pont által reprezentált értéktartományon belül essék.

Az iterációs eljárás tehát a következő lépésekből áll:

- a) Momentum módszerrel kiszámoljuk a becsléspontosságu lecsengési időt. Ezt választjuk  $\tau_2$ -nek. A (12a) egyenlőtlenség teljesülésére a



(13a)  $\tau_1 = 2 \times \tau_2$

(13b)  $\tau_3 = .5 \times \tau_2$

értékeket választjuk. Mindhárom  $\tau$ -hoz kiszámoljuk a megfelelő  $v_1$  értéket.

- b) Megvizsgáljuk, hogy teljesül-e a (12b) reláció? Ha igen, akkor c)-nél folytatjuk.

Ha nem, akkor a követelményt nem teljesítő  $\tau_1$  érték helyett új  $\tau'$  értéket választunk a következő formulák segítségével:

(14a)  $\tau'_1 = 2 \times \tau_1$

(14b)  $\tau'_3 = .5 \times \tau_3$

Az új  $\tau_1$  és  $\tau_3$ -hoz kiszámoljuk a megfelelő  $v_1$  értékeket és visszatérünk a b) ponthoz. (Ebből a ciklusból csak akkor tudunk kilépni, ha a (12b) feltétel teljesül.)

- c) Megfelelő algoritmussal kiválasztjuk a következő próba  $\tau_p$  értékét, amelyre teljesülnie kell a (12c) egyenlőtlenségnek.

Ideális, ha a  $\tau_p$  minél közelebb kerül a valódi  $\tau$ -hoz.

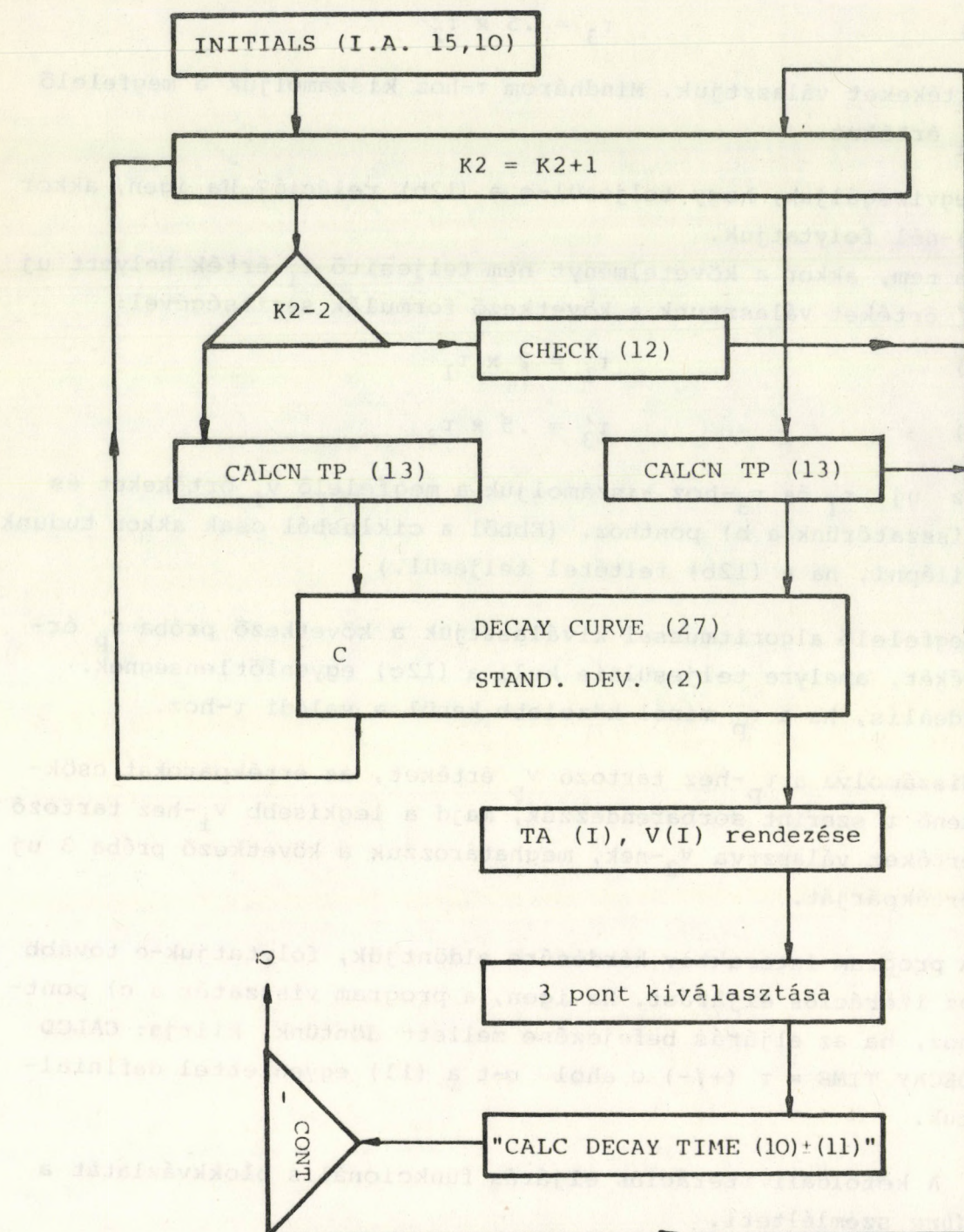
- d) Kiszámolva a  $\tau_p$ -hez tartozó  $v_p$  értéket, az értékpárokat csökkenő  $\tau$  szerint sorbarendezzük, majd a legkisebb  $v_1$ -hez tartozó értéket választva  $v_2$ -nek, meghatározzuk a következő próba 3 új értékpárját.

- e) A program interaktív kérdésére eldöntjük, folytatjuk-e tovább az iterációs eljárást. Ha igen, a program visszatér a c) ponthoz, ha az eljárás befejezése mellett döntünk, kiírja: CALCD DECAY TIME =  $\tau$  (+/-)  $\sigma$  ahol  $\sigma$ -t a (11) egyenlettel definiáltuk.

A kétoldali iterációs eljárás funkcionális blokkvázlatát a 4. ábra szemlélteti.

Az iterációs eljárás lényegét jelentő  $\tau_p$  meghatározására kétféle eljárást használhatunk: (i) a másodfoku parabola és (ii) szilyozott középérték módszert.





4. ábra. A kétoldali közelítés funkcionális blokkvázlata  
(A keretben és zárójelben megadott szám egyenletre hivatkozás)



A kísérleti eredményeket általában módosítás nélkül használjuk. Lehetőség van a kísérleti adattömbök simítására is, de ezt mindig a nem redukált adattömbön végezzük el. A 2-es utasítástömb tartalmazza a 21 pontos ortogonális polinom simitó eljárást.

A simító ablakfüggvény koefficiensei a P tömb elemeiben vannak tárolva. A hosszú ablakfüggvény miatt megköveteljük, hogy a készülékfüggvény félértékszélessége 100 adatpontnál nagyobb legyen.

#### 5.21 A másodfoku közelítés módszere

A három kiválasztott adatpárból a legkisebb négyzetek módszere segítségével kiszámoljuk (11) a három pontra illeszkedő másodfoku egyenlet A, B és C együtthatóit

$$(15) \quad Y = A \cdot t^2 + B \cdot t + C$$

Az így nyert másodfoku egyenlet szélső értéke a keresett valódi  $\tau$  egy jobb  $\tau_p$  közelítését adja:

$$(16) \quad \frac{B}{2 \cdot A} = \tau_p$$

A polinóm fittelés algoritmusát és számítástechnikai vonatkozásait a [11] közleményben tárgyaltuk.

A (16) egyenlettel kiszámított  $\tau_p$  értéket a program megvizsgálja, hogy nem esik-e ki a három pont által reprezentált érték-tartományból (12c reláció teljesülése). Ha ez a helyzet áll elő, akkor a DEC-IT-2ND-nak nevezett program befejezi futását. Mivel tapasztalataink szerint ez már néhány iterációs ciklus után bekövetkezik, mindenképpen a következő pontban ismertetett eljárást tartjuk célravezetőbbnek.

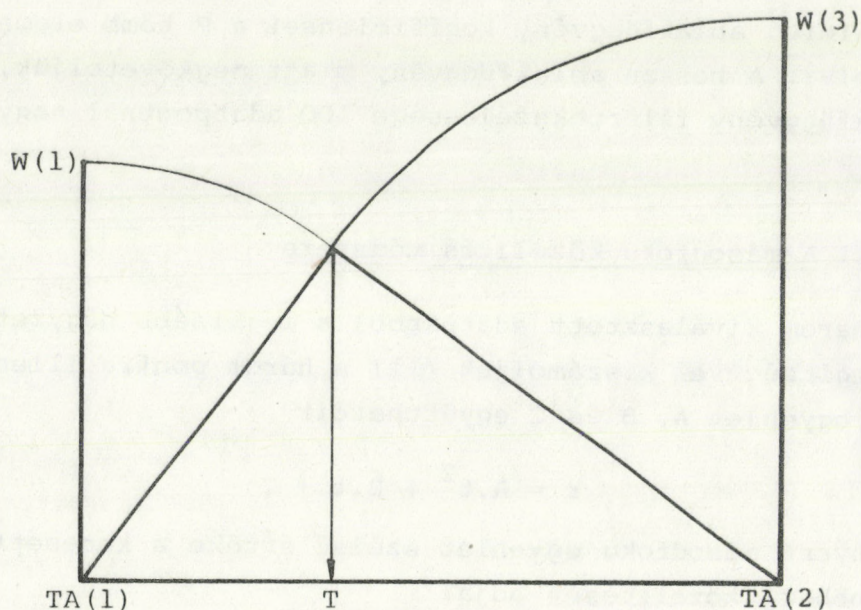
#### 5.22 A súlyozott középértékek módszere

Ha feltételezzük, hogy a három adatpár parabolával közelíthető, akkor a parabola minimuma a  $\tau_1$  és  $\tau_3$  között, a  $v_1$  és  $v_3$  ordinátákkal arányos távolságban helyezkedik el. Az eljárás geometriai interpretációját az 5. ábra szemlélteti. A  $\tau_p$  kiszámításának matematikai megfogalmazása pedig:

$$(17) \quad \tau_p = \tau_3 + v_3 (\tau_1 - \tau_3) / (v_1 + v_3)$$



A (17) egyenlet algoritmusát FOKAL nyelven is egyszerűen megfogalmazhatjuk.



5. ábra. A súlyozott középérték módszer szemléltetése

A DEC-IT-TWO FOKAL nyelvű program iterációs (8-as) utasítás-tömbje:

```

08.10 O L;T !!!"DECAY TIME ITERATION";
08.15 S T=2*T;F K2=1,3;S T=T/2;D 8.4,8.36;
08.16 S K2=3
08.20 I (V(2)-V(1))8.23;S K2=1;S T=2*TA(1);D 8.4,8.36;G 8.2
08.23 I (V(2)-V(3))8.3;S K2=3;S TA(1)=TA(2);S TA(2)=TA(3);D 8.26
08.26 S T=TA(3)/2;S TA(3)=T;D 8.4;G 8.2
08.30 S K2=K2+1;D 8.33,8.4,8.5;G 8.84
08.33 S T=TA(3)+V(3)*(TA(1)-TA(3))/(V(1)+V(3))
08.36 S TA(K2)=T;S V(K2)=SD
08.40 T !!!X2,"TAU(",K2,")=",%,T;D 4
08.50 T !!!"ORDERED SET!";S B=0;D 8.52,8.60,8.64,8.68;R
08.52 F I=1,4;I (0-B)8.56;I (TA(I)-T)8.54;S Z(I)=TA(I);S W(I)=V(I)
08.54 S B=1;S Z(I)=T;S W(I)=SD
08.56 S Z(I)=TA(I-1);S W(I)=V(I-1)
08.60 S B=W(1);F I=1,3;I (B-W(I))1.9,1.9;S B=W(I);S G=I
08.62 I (V(2)-V(3))8.2;S K2=3;S TA(1)=TA(2);S TA(2)=TA(3);D 8.64
08.64 F I=1,3;S TA(I)=Z(I+G-2);S V(I)=W(I+G-2)
08.68 F I=1,3;T !!!"TAU(",X1,I,")=",%,TA(I)," SD=",V(I)
08.84 O T;T !!!"ITERATED DECAY TIME=",TA(2),"(+/-)",(TA(1)-TA(3))/2
08.86 A !!!"CONTINUE? "K1;O L;I (K1-1)8.95,8.3,8.3
08.95 T !!!"CALC DECAY TIME=",%,TA(2),"(+/-)",(TA(1)-TA(3))/2;R

```



### 5.3 Az iterációs algoritmusok értékelése

Az előző pontokban tárgyalt négy iterációs eljárást célszerű a hatékonyság (= időigényesség) szempontjából értékelni. Az egyes iterációs lépések legidőigényesebb eljárása (a szimuláció). lényegében mindegyik eljárásban azonos, ezért a hatékonyság legfontosabb mérőszáma a végeredmény eléréséhez szükséges iterációk száma.

Mintapéldaként a Demas és Crosby által közölt [3] mérési eredményeket használtuk fel és a  $\tau$  értékét  $\pm 0,2\%$  ( $\sigma < |0,01|$ ) pontossággal határoztuk meg minden esetben (amikor ezt a módszer lehetővé tette). Az eredményeket az 1., 2., 3. és 4. táblázat tartalmazza.

Az eredmények alapján a következő megállapításokat tehetjük:

- a) Az egyoldali eljárások hatékonysága egyenértékűnek tekinthető így csak a nagy  $\tau$ -k oldaláról való közelítést ismertettük.
- b) A 2ND eljárás általában csak néhány iterációs lépésig alkalmazható (a Demas-Crosby adatsoron a 8. lépésig, mert a kerekítésből eredő hibák következtében a parabola minimuma kiesik a három pont által reprezentált értéktartományból).
- c) A kétoldali közelítések közül ezért a súlyozott közép (TWO) eljárást célszerű használni, amelynek konvergencia sebessége kb. kétszerese az egyoldali eljárásnak.

### 6. A NORMÁLÁSI FAKTOR (AMPLITUDÓFAKTOR) MEGHATÁROZÁSA

Az iterációs eljárás csak a lecsengési idő meghatározására alkalmas egyparaméteres módszer. A fittelés jóságának meghatározására azonban megfelelő normálást kell alkalmaznunk, hogy az egyes adatpontok eltérés-négyzetösszegét képezhessük. Erre szolgál a normálási faktor, amelynek helyes megválasztását megnehezíti a mérési eredményekre szuperponálódó zaj.

A normálási faktor kiszámítására célszerű egy pontból kiindulni. Ennek az az elvi alapja, hogy ha a számított lecsengési görbe alakja azonos a kísérletiével, akkor egy ponton azonosítva a két görbét, a két görbe fedésbe kerül egymással. A legalkalma-



1. táblázat

A Hundley [2] módszerrel nyert eredmények

Lépésszám	1. ciklus	2. ciklus	3. ciklus	4. ciklus
<u>DT = 1</u>				
1	10			
2	9			
3	8			
4	7			
5	6	<u>DT = .2</u>		
6	5	5.8		
7	4	5.6		
8		5.4		
9		5.2	<u>DT = 0.4</u>	
10		5.0	5.16	
11		4.8	5.12	<u>DT = .008</u>
12			5.08	5.112
13			5.04	5.104
14				5.096
15				5.088
16				5.080
$\tau =$	5	5.0	5.08	5.088
$\pm \sigma =$	1	0.2	0.04	0.008



2. táblázat

A DEC-IT-ONE programmal nyert eredmények

Iterációszám	1. ciklus	2. ciklus	3. ciklus	4. ciklus	5. ciklus	6. ciklus	7. ciklus
	<u>DT=3.38</u>						
1	8.463						
2	6.770						
3	5.642	<u>DT=0.609</u>	<u>DT=0.366</u>				
4	4.795	5.032	5.276	<u>DT=0.122</u>	<u>DT=0.0731</u>		
5	4.118	4.728	5.093	5.154	5.203		
6			4.971	5.093	5.166		
7					5.142	<u>DT=0.013</u>	<u>DT=0.079</u>
8					5.124	5.129	5.134
9					5.109	5.122	5.130
10							5.126
11							5.126
	4.795	5.032	5.093	5.154	5.124	5.129	5.128
	± 0.762	± 0.457	± 0.152	± 0.091	± 0.016	± 0.010	± 0.002



3. táblázat

A DEC-IT-TWO eljárással nyert eredmények

Iterációszám	$\tau_1$	$\tau_2$	$\tau_3$	(+/-) $\sigma$
0	8.463	4.231	2.116	2.1380
1	8.463	5.570	4.231	2.1160
2	5.570	5.278	4.231	0.6690
3	5.278	5.110	4.231	0.5230
4	5.278	5.110	5.030	0.1240
5	5.150	5.110	5.030	0.0601
6	5.150	5.110	5.091	0.0290
7	5.150	5.120	5.110	0.0198
8	5.150	5.130	5.120	0.0144
9	5.135	5.130	5.120	0.0072
10	5.130	5.128	5.120	0.0045

$v_i =$             10.7986        10.7986        10.8036

4. táblázat

A DEC-IT-2ND eljárással nyert eredmények

Iterációszám	$\tau_1$	$\tau_2$	$\tau_3$	(+/-) $\sigma$
3	8.550	4.275	2.137	3.207
4	8.550	5.450	4.275	2.619
5	5.510	5.450	4.275	.618
<hr/>				
6	5.450	5.171	4.275	.588
7	5.171	5.046	4.275	.450
8	5.175	5.046	5.028	.074

$v_i =$             12.879        12.539        12.663



sabbnak látszik erre a kísérleti görbe maximumhelye. A mindig jelen levő zaj azonban teljesen elronthatja normálási eljárásunkat, ezért a zajok kiküszöbölésére kell törekedni. Erre két módszer áll rendelkezésünkre: (i) Az átlag normálási faktor és a (ii) simított maximumhely eljárás.

a) Az  $\overline{NF}$  átlag normálási faktor

A kísérleti görbe maximumhelye környezetében kijelölünk 5-10 pontot, amelynek mindegyikére kiszámoljuk a normálási faktort. Ezeket összegezzük és az átlagértékét képezzük

$$(18) \quad \overline{NF} = \left( \frac{L}{\sum_{I=K} D(I)/C(I)} \right) / (L-K)$$

ahol I az adatpontok számát jelzi, L a kezdő-, K az utolsó figyelembe vett csatornaszám.

Az eljárásnak az a hibája, hogy olyankor amikor a próba  $\tau_p$  értéke eltér a valódi lecsengési időtől, akkor információtartalomban különböző pontokat használunk fel a normálási faktor számítására. Így minél távolabb vagyunk a jó fitteléstől, annál rosszabb a nyert fittelési eredmény. Természetesen jó fittelésnél ez a probléma eltűnik, ezért az eljárás használható. Az átlag normálási faktor használata esetén a felhasznált adatpontok számának négyzetgyökével csökken a normálási faktorra szuperponálódó S kísérleti zaj értéke:

$$(19) \quad S = S_0 / (L-K)^{1/2}$$

b) Az  $NF^+$  simított normálási faktor

A kiválasztott pont (rendszerint maximumhely) mindkét oldalán kiválasztunk 3-10 pontot, s az így nyert páratlan számú elemre illesztünk egy megfelelő fokszámu ortogonális polinómot és meghatározzuk a kiválasztott pont simított értékét. Az eljárás eredményességének az a feltétele, hogy a kiválasztott pontok az adott fokszámu polinómmal jól fittelhetők legyenek (egyébként konvolúciós torzítás következtében a simított érték alacsonyabb lesz, mint a zajmentes kísérleti érték).



Ezután a simitott függvény megfelelő pontját osztva a simitott értékkel nyerjük a  $NF^+$  értékét.

Előnye az eljárásnak az, hogy

- a simitást csak egyszer kell elvégezni (a beolvasási munkafázisban), utána csak egy osztás szükséges a normálási faktorhoz;
- jó normálási faktor esetén az illesztési pont előtti és utáni eltérésösszegek előjele ellentétes (azonos előjelű eltérésösszegek rossz normálási faktorra utalnak).

## 7. A KONVOLUCIÓS INTEGRÁL SZÁMITÁSA

Az (1) integrálegyenlet megoldására kipróbáltuk a különböző numerikus eljárásokat valamint a [11] közölt rekurziós módszert. Tapasztalatainkat az alábbiakban foglalhatjuk össze.

### 7.1 A konvolúciós integrál meghatározása numerikus módszerekkel

A relative kevés adatpont miatt az egyszerűbb integrálközelítések nem adnak jó eredményt. Példafeladatunkat (lásd a 10. pontban) megoldottuk a rendelkezésünkre álló numerikus módszerekkel:

a) téglány szabály

$$(20) \quad S = H \times \sum_i Y_i$$

b) trapéz szabály

$$(21) \quad S = H \times \left( \frac{Y_0 + Y_p}{2} + \sum_{i=1}^{n-1} Y_i \right)$$

c) a módosított Simpson szabállyal (SIMPSON-SPECIAL), amelynek algoritmusát páratlan számú elemre az ismert formulával fogalmazhatjuk meg:

$$(22) \quad S = \frac{H}{3} \left[ Y_0 + Y_n + \sum_{i=1}^{n-1} 4(Y_1 + Y_3 + \dots + Y_{n-1}) + \sum_{i=2}^{n-2} 2(Y_2 + Y_4 + \dots + Y_{n-2}) \right]$$



páros számú elemre azonban nem használható. Javítására a következő módosítást végeztük [12]

egy elem esetén:

$$(23) \quad S = 0$$

két elem esetén a trapéz formulát használjuk:

$$(24) \quad S = (Y_0 + Y_1) / 2$$

kettőnél nagyobb számú páros elemre:

$$(25) \quad S = \frac{H}{3} \left[ Y_0 + Y_{n-1} + \sum_{i=1}^{n-2} 4(Y_1 + Y_3 + \dots + Y_{n-2}) + \sum_{i=2}^{n-3} 2(Y_2 + Y_4 + \dots + Y_{n-3}) \right] + (Y_n + Y_{n-1}) / 2$$

Mindhárom integrálási eljárással futtatuk próbafeladatunkat a DEC-IT-ONE eljárás segítségével. Az iterációt addig folytattuk, amíg a (11) egyenletben definiált  $\sigma$  paraméter értékére teljesül a

$$(26) \quad \sigma \leq 0,01$$

kritérium.

Az eredményeket a 5. táblázat tartalmazza:

5. táblázat

Integrálási módszerek hatása  
a lecsengési idő értékére

Módszer	$\tau$	$\sigma$
Téglány	5,65	0,009
Trapéz	5,13	0,006
Simpson	5,09	0,004

A 5. táblázat és a futtatási tapasztalatok alapján a következő megállapítások tehetők:



- a) Az alkalmazott integrálási eljárás jelentősen kihat a végeredményre. A numerikus módszerek közül mindenképpen a speciális Simpson eljárás tekinthető a legpontosabbnak.
- b) A kívánt kritérium a téglány és trapézközelítés esetén több iterációs lépés után teljesül csak, ezért ezen módszerek gyorsasága nem mindig jelent számottevő időcsökkenést a Simpson eljáráshoz képest.

## 7.2 A konvolúciós integrál meghatározása rekurziós formulával

[11] szerint a konvolúciós integrál a következő rekurziós formulával számolható

$$(27) \quad \sum_{k=1}^n C_{i+1}^k = (C_i^k + 0.5 \cdot H \cdot A_k \cdot K_i) \exp(-H/\tau_k) + 0.5 \cdot H \cdot A_k \cdot K_{i+1}$$

ahol  $n$  az exp. komponensek száma (az iterációs eljárások esetén  $n=1$ ),

$C_i$  és  $K_i$  a számított függvény és készülékfüggvény  $i$ -edik pontja,

$C_0$  és  $K_0$  nullával egyenlő,

$A_k$  az amplitudófaktor.

A rekurziós formulával számított konvoluta pontok jó közelítéssel egyeznek a Simpson integrálással nyert értékekkel, a számolási idő pedig igen nagy mértékben csökken. (A konvolúciós integrál képzése 40 pontra Simpson integrálással kb. 2 perc, míg a rekurziós formulával 8 sec.)

Ezért az a végső következtetés tehető, hogy a konvolúciós integrál számítását mindenképpen a rekurziós eljárással célszerű végrehajtani. A közölt programok ezt az eljárást alkalmazzák.

A 4. szubrutin végzi a konvolúciós integrál rekurziós formulával való számolását és az eltérésnégyzetösszeg képzését, az 5. egység (Simpson Special) pedig a momentumszámítást numerikus eljárással oldja meg.



## 8. A PROGRAM

A DEC-IT- program mindkét változata (a DEC-IT-ONE, és a DEC-IT-TWO) a 8-as utasítástömböt foglalja el. Ezeket az eljárásokat tartalmazó programrészeket az előző pontokban már ismertettük. A következő FOKAL nyelvű programlistában nem szerepelnek annak ellenére, hogy az 5.3 pontban egyértelműen kimutattuk, hogy a DEC-IT-TWO eljárást célszerű használni. A program blokkvázlatán természetesen feltüntettük az iterációs eljárást is, amely mindkét eljárást reprezentálja. A program egyéb részei általános számítás-technikai funkciókat látnak el (adatbevitel, normálás, szimuláció, adatkiadás stb.), amelynek a legtöbb DEC-IT- programnál használhatók. Természetesen különbség van a két iterációs program első soraiban is, ahol a program neve és funkcióját értelmező commentek vannak elhelyezve.

A DEC-IT- programok funkcionális blokkvázlatát a 6. ábrán tüntettük fel.

A program 1-es utasítástömbjét MASTERként használjuk. Ez tartalmazza a programra vonatkozó általános "comment"-eket, a memória mezőkre osztását, a csatornaszélesség beolvasását, a kijelölt memóriamező törlését és a futtatás vezérlését.

A memóriamezők 512 elemes tömböket képeznek. Ezek kezdőcímeit a 6. táblázat tartalmazza.

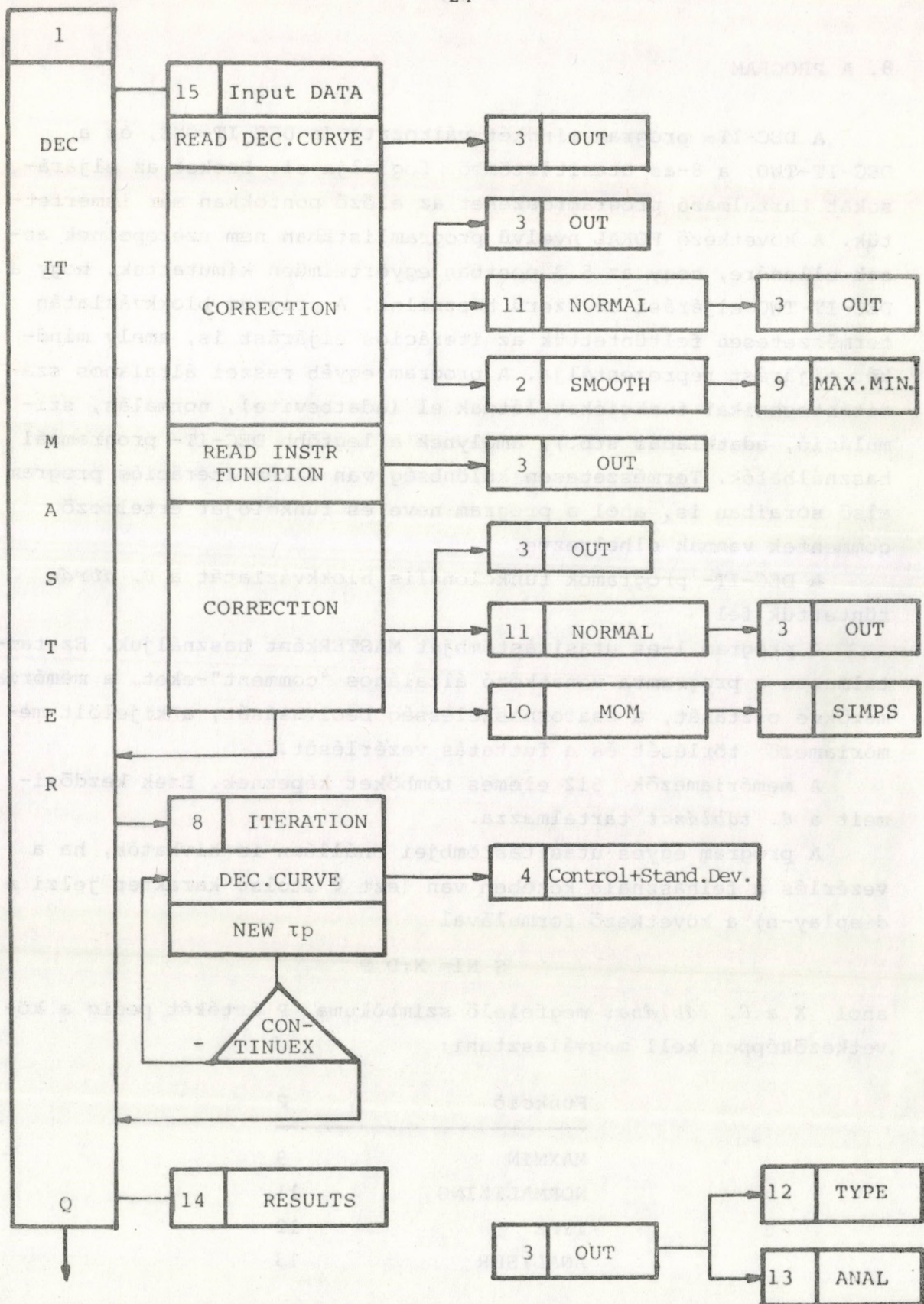
A program egyes utasítástömbjei önállóan is hívhatók, ha a vezérlés a felhasználó kezében van (ezt @ utolsó karakter jelzi a display-n) a következő formulával

$$S N1 = X; D P$$

ahol X a 6. táblázat megfelelő szimbóluma, P értékét pedig a következőképpen kell megválasztani:

Funkció	P
MAXMIN	9
NORMALIZING	11
TYPE	12
ANALYSER	13





6. dbra. A DEC-IT-programok szubrutinszerkezete



6. táblázat

A memóriamezők és kezdőcímei

A mező rendeltetése	X-szimbóluma	Kezdőcime
Kísérleti lecsengési görbe	D	0
Dekonvoluta	G	512
Konvoluta (számított lecsengési görbe)	C	1024
Készülék függvény	K	1536
Átmeneti tár	R	2048

Egyszerűen "D S"-sel hívhatók a következő funkciók:

Funkció	S
MOMENTUM	10
RESULTS	14



# 9. A DEC-IT- PROGRAMLISTA

A listából hiányzik az iterációs (8-as) utasítástömb és az iterációs eljárást azonosító sorok az 1-es utasítástömbből. Ezeket a megfelelő előző pontokban ismertettük.

```
01.20 S LD=512;S LC=1024;S LK=1536;S LR=2048;S NO=0;S YES=1
01.30 O 1;O X;A !! "CHANNEL WIDTH=",CW;O L;D 1.65;D 15,8,14;Q
01.60 1 ! "(CLEAR FNEW);F I=N1,N1+TM;S H=FNEW(I,0)
01.65 T ! "CHANNEL WIDTH=",X5.2,CW
01.90 R

02.05 C SMOOTHING
02.10 S N1=0;D 9.9;S SN=NU;D 2.8,2.3,2.4;R
02.30 S SP=FNEW(SN)*P(0);F I=1,3;S SP=SP+(FNEW(SN+I)+FNEW(SN-I))*P(I)
02.40 T ! "FITTED MAXIMUM=",X,SP
02.50 S P(0)=.118974;S P(1)=.116772;S P(2)=.110128;S P(3)=.990712E-1
02.60 S P(4)=.835913E-1;S P(5)=.636886E-1;S P(6)=.393631E-1
02.70 S P(7)=.106148E-1;S P(8)=.225564E-1;S P(9)=.601504E-1
02.80 S P(0)=.567099;S P(1)=.324675;S P(2)=.12987;S P(3)=.021645
02.90 S SP=FNEW(SN)*P(0);T !X,SN,SP,FNEW(SN),P(0)

03.10 O T;1 ! "DATA OUT?";O K;A K2;I (K2-1)1.9;D 12,13;R

04.05 C CONVUL
04.10 S H=FNEW(LC,0);S EA=0.5*CW;S EX=FEXP(-CW/T)
04.20 F 1=0,1M-1;S A=(FNEW(LC+I)+EA*FNEW(LK+I))*EX+EA*FNEW(LK+I);D 4.3
04.25 S C=SP/FNEW(LC+SN);F I=0,1M;S H=FNEW(LC+I,C*FNEW(LC+I))
04.26 T ! X,"AMPL. FACTOR=",C;S G=0;S SR=0;S SA=0;D 4.5,4.6,4.7;R
04.30 S H=FNEW(LC+1+1,A)
04.50 F I=0,1M;S B=FNEW(LR+J,FNEW(LC+I)-FNEW(I));D 4.55
04.55 S H=FNEW(LR+I,B);S G=G+B*B;I (I-SN)4.8,1.9;S SA=SA+B
04.60 S SD=FSQT(G/1M);T ! "AV.DEV.AT",X3,K1," TH ITER.=",X,SD
04.70 T ! "SUM BEFORE=",SB," SUM AFTER=",SA," ALL=",SA+SB
04.80 S SB=SB+B

05.05 C SIMPSON - SPECIAL. R1,R2 AND H ARE EXTERNAL PARAMETERS
05.10 S S=0;S B=0;S G=0;S TR=0;I (R2-1-R1)1.9,5.2,5.3
05.15 S S=TR+(S+4*B+2*G)*CW/3;R;T ! "S=",S
05.20 S S=.5*FNEW(R2)
05.30 S L=-1;S RR=R2-R1;D 5.35,5.5,5.7,5.15;R
05.35 I (F1R(RR/2)*2-RR)5.4;S L=0;G 5.4
05.40 S S=FNEW(R1)+FNEW(R2+L);I (R2+L-R1-1)1.9,5.45,5.45
05.45 S B=B+FNEW(R1+1);R;T ! "B",B,FNEW(R1+1)
05.50 F I2=R1+2,2,R2-1+L;I (R2+L-I2-2)1.9;D 5.6
05.60 S G=G+FNEW(I2);S B=B+FNEW(I2+1);R;T ! "GB",G,B,FNEW(I2),FNEW(I2+1)
05.70 I (-L)1.9,1.9;S 1R=(FNEW(R2)+FNEW(R2-1))*CW/2;R;T ! "TR",TR,FNEW(R2)

07.10 O P,C;F I=0,100;T 0
07.20 W A;
07.30 D 7.1;O 1,1;R
07.40 R
```



```

09.05 C MAXMIN
09.10 D 9.7,9.3,9.2,9.5;R
09.20 O L;I !X8.5,"MAX:",SU," AT ",X5,NU," MIN:",X8.5,SL," AT ",X5,NL
09.30 S NL=N1;S SL=SU;F I=N1,N1+TM;I (FNEW(I)-SU)9.4;S SU=FNEW(I);S NU=I
09.40 I (SL-FNEW(I))1.9;S SL=FNEW(I);S NL=I
09.50 O I;A "RORD MAX.=",G;O L;I!"RORD MAX.=",X,G;S G=G/(SU-SL);D 9.6
09.60 F I=N1,N1+TM;S H=FNEW(I),(FNEW(I)-SL)*G)
09.70 S SU=FNEW(N1);S NU=N1;
09.90 D 9.7,9.3,9.2;R

10.10 O L;I ! "MOMENTUM APPROXIMATION";S R1=LR;D 10.2,10.3,10.5;R
10.20 S R2=R1+TM;F L1=0,1;S B1=LK;S G1=L1;D 10.4,5,10.25
10.25 S MK(L1)=S;T ! "MK(",X1,L1,")=",X,MK(L1)
10.30 F L1=0,1;S B1=0;S G1=L1;D 10.4,5;S MD(L1)=S;D 10.35
10.35 T ! "MD(",X1,L1,")=",X,MD(L1)
10.40 F I=0,TM;S H=FNEW(LR+I,FNEW(B1+I)*(I*CW)^L1)
10.50 S T=(MD(1)/MD(0))-(MK(1)/MK(0))

11.10 O I,K;A ! "NORMALISING? ",K;I (K1-1)1.9;O L;I!"NORMALISING";D 11.2,3.1;R
11.20 O T;A ! "MAXIMUM(1)-INTEGRAL(2)",K;I (K1-1)1.9,9.1,11.5
11.50 S R1=N1;S R2=N1+TM;D 5,11.94;O K;O T;A ! "RORD INTEGRAL=",D;D 11.92,9.6
11.92 O L;I ! "RORD INTEGRAL=",X,D;S G=D/S;S SL=0
11.94 O L;I ! "INTEGRAL=",X,S

12.10 O T;A ! "TYPE? ",K;I (K1-1)1.9;O L;S L=67;D 12.2,12.5,12.3;R
12.20 T !&7,"I",&14,"X",;F I=0,4;T X1," ",I
12.30 F I=N1,5,N1+TM;T !X6,I,I-N1," : ";D 12.4
12.40 F J=0,4;I (N1+TM-I-J)1.9;T X8.3,FNEW(I+J)
12.50 T !;F I=0,L;T "--"

13.10 O K;O I;A ! "ANALYSER? ",K;I (K3-1)1.9
13.12 S P(0)=48;S P(1)=177;S P(2)=178;S P(3)=51;S P(4)=180
13.14 S P(5)=53;S P(6)=54;S P(7)=183;S P(8)=184;S P(9)=57;S P(10)=58
13.15 D 7.1,13.2,7.3;R
13.20 F I=N1,N1+TM;S C=100000;S B=FITR(FNEW(I));D 13.5,13.4;T P(10)
13.40 F J=1,5;S C=C/10;S D=FITR(B/C);T P(D);S B=B-C*D
13.50 I (-B)1.9;S B=0

14.10 O T;A ! "RESULTS? ",K;I (K1-1)1.9;O L;S L=65;D 14.2,12.5,14.3;R
14.20 T !&3,"TIME",&12,"K(T)",&21,"D(T)",&33,"C(T)",&43;D 14.25
14.25 T ! "D(T)-C(T)",&60,"REL.SD"
14.30 F I=0,TM;D 14.4
14.40 T !X5,I,X8,FNEW(LK+I),FNEW(I),X10.3,&26,FNEW(LC+I);D 14.45
14.45 T !&40,X10.5,FNEW(LR+1),&54,FNEW(LR+I)/V(2)

15.10 O X;O T;A ! "READ DATA! ",NR OF DATA=",TM;D 15.92;S TM=TM-1
15.15 S L=0;S N1=0;D 15.9,2,15.6,15.3;R
15.20 O K;O T;A ! "TAPE IN",K;O R;S 186;F I=0,TM;A B; S H=FNEW(L+I,B)
15.30 D 10;O L;I !X,"ASSESSED DECAY TIME=",T;S T=2*T
15.40 O K;O T;A ! "CORRECTION? ",K;I (K1-1)1.9;O L;I ! "CORRECTION";D 15.5
15.50 O I;A ! "I=",I," Y=",B;I (I)1.9;S H=FNEW(N1+I,B);G 15.5
15.60 O T;A ! "READ INSTR.FUNC";O L;D 15.95;S L=LK;S N1=L;D 15.9
15.70 I (1-K1)15.2;F I=0,TM;T !X5,I,&6,"=";A B;S H=FNEW(L+I,B)
15.90 O T;A ! "TI(1) - TR(2)? ",K;D 15.7,3.1,15.4,3.1,11
15.92 O L;I ! "NUMBER OF DATA=",X4.0,TM;T ! "DECAY CURVE"
15.95 T ! "INSTRUMENT FUNCTION"

```



## 10. A DEC-IT- PROGRAMOK HASZNÁLATA

### A) Indítás

A G(R) leütése után a program kiírja a nevét a mozaik printer  
DEC-IT...

amely lehet ONE vagy TWO

### B) Input információk és adatok

#### B/1 A program a display-n kiírja:

CHANNEL WIDTH=

és bekéri a két szomszédos csatorna közti időkülönbség értékét. A csatornaszélesség értéke a mozaikprinteren is megjelenik.

#### B/2 A display-n megjelenő READ DATA felirat arról informálja a felhasználót, hogy az adattömbök beolvasása következik.

A NUMBER OF DATA= kérdésre a válasz az adattömb nagysága.  
(Az adattömb nagyságát a mozaikprinter is kiírja.)

#### B/3 A mozaikprinteren megjelenő DECAY CURVE felirat jelzi, hogy a lecsengési görbe beolvasása következik.

#### B/4 Periféria választás

A display-n megjelenő

TT/1/-TR/2/

kérdésre a válasz 1(R) vagy 2(R) aszerint, hogy a display keyboard-járól vagy a gyorsolvasóval akarjuk bevinni az adatokat.

##### a) 1-es válasz esetén a display-n kiírja:

O.= és várja a O.-dik adatot. Beütés és "R" terminátor után az új kiírás:

1.= stb.

##### b) 2-es válasz esetén a TAPE IN! felirat figyelmezteti a felhasználót, hogy az adatszalagot helyezze a gyorsolvasóba. A beolvasás "R" terminátor beütésével indítható.

#### B/5 A display-n megjelenő

DATA OUT? kérdésre a válasz YES vagy NO, aszerint, hogy a bevitt tömböt ki akarjuk-e hozni. (Az adatkihozatal részleteit lásd a B/12. pontban.)



- B/6 A display-n megjelenő CORRECTION? felirattal érdeklődik a program, vajon a felhasználó akar-e javítást végrehajtani a bevitt adattömbben. YES válasz esetén a mozaikprinteren is megjelenik a CORRECTION felirat, majd a display-n az I= , Y= kérdésekre a javítandó adat sorszámát és helyes értékét kell megadni. (A válaszokat célszerű "," terminátorral zárni, így a két összetartozó adat egy sorba kerül.) A korrekciós ciklusból úgy lehet kilépni, hogy a I= kérdésre negatív számot adunk meg.
- B/7 A DATA OUT? kérdéssel a program ismét lehetőséget biztosít a tömbkihozatalra.
- B/8 A display-n megjelenik a NORMALISING? kérdés, amely az adattömb normálását teszi lehetővé. (A normálás részleteit lásd a B/13 pontban.)
- B/9 Az előző kérdésre adott igenlő válasz esetén a DATA OUT? felirat ismételt tömbkihozatalra biztosít lehetőséget.
- B/10 Fittelés  
A program megkeresi az adattömb maximum és minimumhelyét és a hozzátartozó időkoordinátát, majd kiírja a sornyomtatón:  
MAX...at... MIN...at...  
Beavatkozás nélkül megfitteli a maximumhelyhez tartozó adatot és kiírja a fittelt értéket:  
FITTED MAX=
- B/11 A mozaikprinteren és a display-n megjelenő INSTRUMENT FUNCTION felirat jelzi, hogy a készülékfüggvény beolvasása következik. A program ezután B/4-B/9-ig megismétli az eljárást.
- B/12 Adatkihozatal  
A display-n megjelenő DATA OUT? kérdésre adott igenlő válasz esetén a program megkérdezi:  
TYPE? Igenlő válasz esetén az éppen aktuális tömb kiírásra kerül a mozaikprinteren. (A kiírás olyan formában történik, hogy feltüntetjük az egyes elemek tömbön belüli sorszámát és az FNEW mezőben való elhelyezkedésüket is.)  
Az ANALYSER? kérdésre adott YES válasz esetén az aktuális tömb analízátor kódban lelyukasztásra kerül a gyorslyukasztón.



### B/13 Normálás

Amennyiben a beolvasott adattömb normálását kérjük (a NORMALISING kérdésre adott YES válasz esetén), a NORMALISING felirat a mozaikprinteren is megjelenik, majd a program felteszi a következő kérdést:

MAXIMUM/1/- INTEGRAL/2/- azaz amplitudó vagy területnormálást kérünk-e.

1. Az első esetben a mozaikprinteren megjelenik a

MAX... at... MIN... at... felirat, azaz a normálásra váró adattömb jelenlegi maximum- és minimum értéke és helye.

Ezután a display-n megjelenő RQRD MAX= kérdésre meg kell adni a kívánt maximum értékét. (Ez az érték a mozaikprinteren is megjelenik.)

A program ezután végrehajtja a normálást a következő algoritmus szerint:

$D_{\text{norm}}(t) = (D(t) - D_{\text{min}}) * G$  ahol a G normálási faktor

$$G = \frac{\text{RQRD MAX}}{(D_{\text{max}} - D_{\text{min}})}$$

2. Területnormálás esetén a program kiszámolja a normálásra váró adattömb alatti területet. Az eredményt a mozaikprinteren megjelenő INTEGRAL=... felirattal hozza a felhasználó tudtára. Ezután megkérdezi a kívánt terület értékét:

RQRD INTEGRAL= , ezt az értéket kiírja a mozaikprinteren is, majd az alábbi módon végrehajtja a normálást:

$$D_{\text{norm}}(t) = D(t) \cdot G, \text{ ahol } G = \frac{\text{RQRD INTEGRAL}}{\text{INTEGRAL}}$$

### C) A becsült $\tau$ kiszámítása

Külön beavatkozás nélkül a program a momentum módszerrel kiszámolja a közelítő  $\tau$  értékét, és eredményeit a következő formában közli a mozaikprinteren a felhasználóval:

MOMENTUM APPROXIMATION

MK/O/=...

MK/1/=...

MD/O/=...



MD/1/=...

ASSESSED DECAY TIME=...

A program ezután rátér az iterációs ciklusra.

D/1 Iteráció egyoldali közelítés esetén (DEC-IT- ONE)

a) A program a DECAY TIME ITERATION felirattal közli az iterációs ciklus kezdetét.

b) A lépésköz kiírása  $DT=...$  formában történik.

c) A program közli az iteráció sorszámát és a kezdőparamétereket:

...TH ITERATION

TAU/I/=...

AMP.FACTOR=...

d) A program a 7.2. pontban ismerttetett rekurziós eljárással kiszámolja a konvolúciós integrált. A számítás eredményét a következő formában közli:

kiírja az egy pontra számolt átlagos eltérésnégyzetösszeget:

AV. DEV. AT ...TH ITERATION=...

A maximumhely előtt és után számolt, valamint az összes pontra számolt eltérésösszeget:

SUM BEFORE... SUM AFTER... ALL...

Az iterációhoz felhasznált utolsó három  $\tau$  értékét és a hozzájuk tartozó standard deviationont a következő formában közli:

ORDERED SET

Tau/1/=... SD/1/=...

.

.

.

TAU/3/=... SD/3/=...

e) Az iterációs ciklus vége

A ciklus akkor ér véget, amikor az új  $\tau$ -hoz tartozó SD nagyobb lesz mint az előző. A program ezt azzal közli, hogy kiírja a számított  $\tau$  értékét

CALC DECAY TIME=.../+/-/...

Közli az új lépésköz értékét

DT=...



majd a display-n felteszi a  
CONTINUE? kérdést. Igenlő válasz esetén a TAU és SD tömb  
elemeit 2-vel nagyobb indexű helyre shifteli és az  
ORDERED SET felírás után kiírja az új tömböket.  
Ezután új iterációs ciklusba kezd és a c). ponttól megis-  
métli az eljárást.

A CONTINUE? kérdésre adott nemleges válasz esetén a prog-  
ram felteszi a RESULTS? kérdést, azaz a felhasználó ki-  
vánja-e az eredmények részletes kiírását. Nemleges válasz  
esetén a futás befejeződik, ellenkező esetben a program  
fejlécezt készít és kitölti a következő táblázatot.

TIME	K(t)	D(t)	C(t)	D(t)-C(t)	REL SD

(A REL SD jelentése  $\frac{D(t)-C(t)}{SD}$  ahol SD a véglegesen elfo-  
gadott  $\tau$ -hoz tartozó standard deviation.)

#### D/2 Iteráció kétoldali közelítés esetén (DEC-IT-TWO)

A program a mozaikprinterén felírással jelzi, hogy a  
DECAY TIME ITERATION eljárás következik.

- a) A momentum módszerrel nyert közelítő  $\tau$ -ból meghatározza  
az első három  $\tau(i)$ ,  $v(i)$  értékpárt az 5.2 fejezetben le-  
írtak alapján.

A számolás eredményeit a következő formában közli:

AMPL. FACTOR=...

AV. DEV. AT I-TH ITERATION=...

SUM BEFORE=... SUM AFTER=... ALL=...

- b) A három értékpár alapján új  $\tau$ -t határoz meg, a közelítés  
eredményét az a) pontban leírt formában jelenti meg a  
mozaikprinterén, majd megállapítja a további közelítés  
alapjául szolgáló három új értékpárt:

ORDERED SET

TAU/1/=... SD/1/=...

⋮

TAU/3/=... SD/3/=...



c) Az iterációs ciklus vége

A display-n kiírja:

ITERATED DECAY TIME=  $\tau/2/\pm\sigma$

A CONTINUE? kérdésre adott igenlő válasz esetén visszatér

a b) ponthoz nemleges válasz esetén a mozaikprinteren is kiírja az eredményt:

CALC DECAY TIME=...( $\pm$ )...

A RESULTS? kérdésre adott igenlő válasz esetén a mozaikprinteren elkészíti a D1/e pontban ismertetett táblázatot.

## 11. FUTTATÁSI MINTAPÉLDÁK

A DEC-IT- eljárásokat a Demas-Crosby által közölt mérési eredmények értékelése alapján hasonlítottuk össze. Az alábbiakban a legmegfelelőbbnek bizonyult DEC-IT-ONE és DEC-IT-TWO programokkal nyert futtatási példákat közöljük.

### 11.1 DEC-IT-ONE futtatási mintapélda

DEC-IT-ONE

CHANNEL WIDTH= 1.00

NUMBER OF DATA= 40

DECAY CURVE

	1	X	0	1	2	3	4
0	0 :	0.000	0.000	4.000	24.000	65.000	
5	5 :	138.000	222.000	322.000	438.000	549.000	
10	10 :	665.000	760.000	840.000	911.000	955.000	
15	15 :	980.000	998.000	1000.000	981.000	963.000	
20	20 :	929.000	892.000	849.000	793.000	745.000	
25	25 :	682.000	628.000	574.000	521.000	464.000	
30	30 :	415.000	367.000	330.000	292.000	257.000	
35	35 :	223.000	196.000	168.000	145.000	121.000	

MAX: 1000.0000 AT 17 MIN: 0.00000 AT 0

FITTED MAXIMUM= .998073E+03



# INSTRUMENT FUNCTION

I	X	0	1	2	3	4
1536	0 :	0.000	15.000	41.000	128.000	281.000
1541	5 :	393.000	595.000	691.000	835.000	890.000
1546	10 :	954.000	1000.000	992.000	990.000	964.000
1551	15 :	913.000	873.000	812.000	744.000	672.000
1556	20 :	606.000	533.000	472.000	412.000	362.000
1561	25 :	319.000	270.000	236.000	206.000	172.000
1566	30 :	143.000	122.000	103.000	88.000	73.000
1571	35 :	64.000	54.000	47.000	39.000	34.000

## MOMENTUM APPROXIMATION

MK( 0) = .170908E+05

MK( 1) = .261753E+06

MD( 0) = .203427E+05

MD( 1) = .397631E+06

ASSESSED DECAY TIME = .423127E+01

## DECAY TIME ITERATION

DT = .338502E+01

## 1TH ITERATION

TAU( 1) = .846255E+01

AMPL. FACTOR = .170546E+00

AV.DEV.AT 1 TH ITER. = .121387E+03

SUM BEFORE = -.953603E+03 SUM AFTER = .313807E+04 ALL = .218447E+04

## ORDERED SET:

TA( 1) = .000000E+00 SD( 1) = .000000E+00

TA( 2) = .000000E+00 SD( 2) = .000000E+00

TA( 3) = .846255E+01 SD( 3) = .121387E+03

## 2TH ITERATION

TAU( 2) = .677004E+01

AMPL. FACTOR = .194548E+00

AV.DEV.AT 2 TH ITER. = .644755E+02

SUM BEFORE = -.636856E+03 SUM AFTER = .158810E+04 ALL = .951249E+03



ORDERED SET:

TA( 1)= .000000E+00 SD( 1)= .000000E+00

TA( 2)= .846255E+01 SD( 2)= .121387E+03

TA( 3)= .677004E+01 SD( 3)= .644755E+02

3TH ITERATION

TAU( 3)= .564170E+01

AMPL. FACTOR= .219952E+00

AV.DEV.A1 3 TH ITER.= .234097E+02

SUM BEFORE=-.333823E+03 SUM AFTER= .452380E+03 ALL= .118557E+03

ORDERED SET:

TA( 1)= .846255E+01 SD( 1)= .121387E+03

TA( 2)= .677004E+01 SD( 2)= .644755E+02

TA( 3)= .564170E+01 SD( 3)= .234097E+02

4TH ITERATION

TAU( 4)= .479544E+01

AMPL. FACTOR= .248212E+00

AV.DEV.A1 4 TH ITER.= .176231E+02

SUM BEFORE=-.292141E+02 SUM AFTER=-.439910E+03 ALL=-.469124E+03

ORDERED SET:

TA( 1)= .677004E+01 SD( 1)= .644755E+02

TA( 2)= .564170E+01 SD( 2)= .234097E+02

TA( 3)= .479544E+01 SD( 3)= .176231E+02

5TH ITERATION

TAU( 5)= .411844E+01

AMPL. FACTOR= .280582E+00

AV.DEV.A1 5 TH ITER.= .447768E+02

SUM BEFORE= .284503E+03 SUM AFTER=-.116495E+04 ALL=-.880450E+03

CALC. DECAY TIME= 4.795(+/-) 0.762

H1= .609304E+00



ORDERED SET:

TA( 1)= .846255E+01 SD( 1)= .121387E+03

TA( 2)= .677004E+01 SD( 2)= .644755E+02

TA( 3)= .564170E+01 SD( 3)= .234097E+02

4TH ITERATION

TAU( 4)= .503239E+01

AMPL. FACTOR= .239201E+00

AV.DEV.AT 4 TH ITER.= .115070E+02

SUM BEFORE=-.122973E+03 SUM AFTER=-.187624E+03 ALL=-.310597E+03

ORDERED SET:

TA( 1)= .677004E+01 SD( 1)= .644755E+02

TA( 2)= .564170E+01 SD( 2)= .234097E+02

TA( 3)= .503239E+01 SD( 3)= .115070E+02

5TH ITERATION

TAU( 5)= .472774E+01

AMPL. FACTOR= .250977E+00

AV.DEV.AT 5 TH ITER.= .199760E+02

SUM BEFORE=-.104026E+01 SUM AFTER=-.512232E+03 ALL=-.513272E+03

CALC. DECAY TIME= 5.032(+/-) 0.457

DT= .365583E+00

ORDERED SET:

TA( 1)= .846255E+01 SD( 1)= .121387E+03

TA( 2)= .677004E+01 SD( 2)= .644755E+02

TA( 3)= .564170E+01 SD( 3)= .234097E+02

4TH ITERATION

TAU( 4)= .527611E+01

AMPL. FACTOR= .230891E+00

AV.DEV.AT 4 TH ITER.= .123859E+02

SUM BEFORE=-.212174E+03 SUM AFTER= .701193E+02 ALL=-.142054E+03



ORDERED SET:

TA( 1)= .677004E+01 SD( 1)= .644755E+02

TA( 2)= .564170E+01 SD( 2)= .234097E+02

TA( 3)= .527611E+01 SD( 3)= .123859E+02

5TH ITERATION

TAU( 5)= .509332E+01

AMPL. FACTOR= .237038E+00

AV.DEV.AT 5 TH ITER.= .108951E+02

SUM BEFORE=-.145932E+03 SUM AFTER=-.123001E+03 ALL=-.268933E+03

ORDERED SET:

TA( 1)= .564170E+01 SD( 1)= .234097E+02

TA( 2)= .527611E+01 SD( 2)= .123859E+02

TA( 3)= .509332E+01 SD( 3)= .108951E+02

6TH ITERATION

TAU( 6)= .497146E+01

AMPL. FACTOR= .241424E+00

AV.DEV.AT 6 TH ITER.= .126123E+02

SUM BEFORE=-.995565E+02 SUM AFTER=-.252357E+03 ALL=-.351914E+03

CALC. DECAY TIME= 5.093(+/-) 0.152

DT= .121861E+00

ORDERED SET:

TA( 1)= .677004E+01 SD( 1)= .644755E+02

TA( 2)= .564170E+01 SD( 2)= .234097E+02

TA( 3)= .527611E+01 SD( 3)= .123859E+02

5TH ITERATION

TAU( 5)= .515425E+01

AMPL. FACTOR= .234934E+00

AV.DEV.AT 5 TH ITER.= .108514E+02

SUM BEFORE=-.168441E+03 SUM AFTER=-.585046E+02 ALL=-.226945E+03



ORDERED SET:

TA( 1)= .564170E+01 SD( 1)= .234097E+02

TA( 2)= .527611E+01 SD( 2)= .123859E+02

TA( 3)= .515425E+01 SD( 3)= .108514E+02

6TH ITERATION

TAU( 6)= .509332E+01

AMPL. FACTOR= .237038E+00

AV.DEV.AT 6 TH ITER.= .108950E+02

SUM BEFORE=-.145929E+03 SUM AFTER=-.123004E+03 ALL=-.268933E+03

CALC. DECAY TIME= 5.154(+/-) 0.091

DT= .731171E-01

ORDERED SET:

TA( 1)= .677004E+01 SD( 1)= .644755E+02

TA( 2)= .564170E+01 SD( 2)= .234097E+02

TA( 3)= .527611E+01 SD( 3)= .123859E+02

5TH ITERATION

TAU( 5)= .520300E+01

AMPL. FACTOR= .233290E+00

AV.DEV.AT 5 TH ITER.= .112276E+02

SUM BEFORE=-.186138E+03 SUM AFTER=-.698715E+01 ALL=-.193125E+03

ORDERED SET:

TA( 1)= .564170E+01 SD( 1)= .234097E+02

TA( 2)= .527611E+01 SD( 2)= .123859E+02

TA( 3)= .520300E+01 SD( 3)= .112276E+02

6TH ITERATION

TAU( 6)= .516644E+01

AMPL. FACTOR= .234519E+00

AV.DEV.AT 6 TH ITER.= .109123E+02

SUM BEFORE=-.172891E+03 SUM AFTER=-.456166E+02 ALL=-.218508E+03



ORDERED SET:

TA( 1)= .527611E+01 SD( 1)= .123859E+02

TA( 2)= .520300E+01 SD( 2)= .112276E+02

TA( 3)= .516644E+01 SD( 3)= .109123E+02

7TH ITERATION

TAU( 7)= .514207E+01

AMPL. FACTUR= .235350E+00

AV.DEV.AT 7 TH ITER.= .108135E+02

SUM BEFORE=-.163972E+03 SUM AFTER=-.713990E+02 ALL=-.235371E+03

ORDERED SET:

TA( 1)= .520300E+01 SD( 1)= .112276E+02

TA( 2)= .516644E+01 SD( 2)= .109123E+02

TA( 3)= .514207E+01 SD( 3)= .108135E+02

8TH ITERATION

TAU( 8)= .512378E+01

AMPL. FACTUR= .235979E+00

AV.DEV.AT 8 TH ITER.= .108001E+02

SUM BEFORE=-.157239E+03 SUM AFTER=-.907400E+02 ALL=-.247979E+03

ORDERED SET:

TA( 1)= .516644E+01 SD( 1)= .109123E+02

TA( 2)= .514207E+01 SD( 2)= .108135E+02

TA( 3)= .512378E+01 SD( 3)= .108001E+02

9TH ITERATION

TAU( 9)= .510916E+01

AMPL. FACTUR= .236485E+00

AV.DEV.AT 9 TH ITER.= .108275E+02

SUM BEFORE=-.151823E+03 SUM AFTER=-.106224E+03 ALL=-.258048E+03

CALC. DECAY TIME= 5.124(+/-) 0.016

DT= .131615E-01



ORDERED SET:

TA( 1)= .520300E+01 SD( 1)= .112276E+02

TA( 2)= .516644E+01 SD( 2)= .109123E+02

TA( 3)= .514207E+01 SD( 3)= .108135E+02

8TH ITERATION

TAU( 8)= .512890E+01

AMPL. FACTOR= .235802E+00

AV.DEV.AT 8 TH ITER.= .107986E+02

SUM BEFORE=-.159130E+03 SUM AFTER=-.853203E+02 ALL=-.244450E+03

ORDERED SET:

TA( 1)= .516644E+01 SD( 1)= .109123E+02

TA( 2)= .514207E+01 SD( 2)= .108135E+02

TA( 3)= .512890E+01 SD( 3)= .107986E+02

9TH ITERATION

TAU( 9)= .512232E+01

AMPL. FACTOR= .236029E+00

AV.DEV.AT 9 TH ITER.= .108014E+02

SUM BEFORE=-.156697E+03 SUM AFTER=-.922952E+02 ALL=-.248992E+03

CALC. DECAY TIME= 5.129(+/-) 0.010

DT= .789718E-02

ORDERED SET:

TA( 1)= .520300E+01 SD( 1)= .112276E+02

TA( 2)= .516644E+01 SD( 2)= .109123E+02

TA( 3)= .514207E+01 SD( 3)= .108135E+02

8TH ITERATION

TAU( 8)= .513417E+01

AMPL. FACTOR= .235621E+00

AV.DEV.AT 8 TH ITER.= .108013E+02

SUM BEFORE=-.161069E+03 SUM AFTER=-.797525E+02 ALL=-.240822E+03



ORDERED SET:

TA( 1)= .516644E+01 SD( 1)= .109123E+02

TA( 2)= .514207E+01 SD( 2)= .108135E+02

TA( 3)= .513417E+01 SD( 3)= .108013E+02

9TH ITERATION

TAU( 9)= .513022E+01

AMPL. FACTOR= .235757E+00

AV.DEV.AT 9 TH ITER.= .107988E+02

SUM BEFORE=-.159613E+03 SUM AFTER=-.839311E+02 ALL=-.243544E+03

ORDERED SET:

TA( 1)= .514207E+01 SD( 1)= .108135E+02

TA( 2)= .513417E+01 SD( 2)= .108013E+02

TA( 3)= .513022E+01 SD( 3)= .107988E+02

10TH ITERATION

TAU( 10)= .512758E+01

AMPL. FACTOR= .235848E+00

AV.DEV.AT 10 TH ITER.= .107986E+02

SUM BEFORE=-.158642E+03 SUM AFTER=-.867190E+02 ALL=-.245361E+03

ORDERED SET:

TA( 1)= .513417E+01 SD( 1)= .108013E+02

TA( 2)= .513022E+01 SD( 2)= .107988E+02

TA( 3)= .512758E+01 SD( 3)= .107986E+02

11TH ITERATION

TAU( 11)= .512561E+01

AMPL. FACTOR= .235916E+00

AV.DEV.AT 11 TH ITER.= .107992E+02

SUM BEFORE=-.157912E+03 SUM AFTER=-.888126E+02 ALL=-.246725E+03

CALC. DECAY TIME= 5.128(+/-) 0.002



DT= .184326E-02

TIME	K(T)	D(T)	C(T)	D(T)-C(T)	REL.SD
0	0	0	0.000	0.00000	0.00000
1	15	0	1.769	1.76937	0.16381
2	41	4	7.748	3.74778	0.34697
3	128	24	25.452	11.09800	1.02747
4	281	65	66.509	1.50945	0.13975
5	393	138	128.349	9.65057	0.89346
6	595	222	213.926	8.07440	0.74754
7	691	322	315.262	6.73828	0.62384
8	835	438	424.939	13.06080	1.20919
9	890	549	535.639	13.36060	1.23694
10	954	665	639.606	25.39440	2.35105
11	1000	760	736.781	23.21860	2.14961
12	992	840	820.254	19.74620	1.82813
13	990	911	887.918	23.08140	2.13691
14	964	955	940.329	14.67100	1.35826
15	913	980	974.911	5.08936	0.47118
16	873	998	993.695	4.30505	0.39857
17	812	1000	998.072	1.92773	0.17847
18	744	981	987.733	6.73254	0.62331
19	672	963	964.133	1.13330	0.10492
20	606	929	929.944	0.94385	0.08738
21	533	892	886.798	5.20178	0.48159
22	472	849	837.020	11.98000	1.10912
23	412	793	783.067	9.93274	0.91959
24	362	745	726.956	18.04350	1.67049
25	319	682	670.867	11.13340	1.03075
26	270	628	614.765	13.23470	1.22529
27	236	574	559.842	14.15830	1.31080
28	206	521	507.815	13.18520	1.22070
29	172	464	458.087	5.91266	0.54740
30	143	415	410.453	4.54657	0.42093
31	122	367	365.971	1.02905	0.09527
32	103	330	325.094	4.90643	0.45424
33	88	292	287.848	4.15173	0.38437
34	73	257	253.979	3.02054	0.27965
35	64	223	223.596	0.59641	0.05522
36	54	196	196.546	0.54559	0.05051
37	47	168	172.493	4.49323	0.41599
38	39	145	151.081	6.08109	0.56300
39	34	121	132.098	11.09800	1.02747



# 11.2 DEC-IT-TWO futtatási mintapélda

DEC-IT-TWO

CHANNEL WIDTH= 1.00

NUMBER OF DATA= 40

DECAY CURVE

I	X	0	1	2	3	4
0	0 :	0.000	0.000	4.000	24.000	65.000
5	5 :	138.000	222.000	322.000	438.000	549.000
10	10 :	665.000	760.000	840.000	911.000	955.000
15	15 :	980.000	998.000	1000.000	981.000	963.000
20	20 :	929.000	892.000	849.000	793.000	745.000
25	25 :	682.000	628.000	574.000	521.000	464.000
30	30 :	415.000	367.000	330.000	292.000	257.000
35	35 :	223.000	196.000	168.000	145.000	121.000

MAX: 1000.0000 AT 17 MIN: 0.00000 A1 0

FITTED MAXIMUM= .998073E+03

INSTRUMENT FUNCTION

I	X	0	1	2	3	4
1536	0 :	0.000	15.000	41.000	128.000	281.000
1541	5 :	393.000	595.000	691.000	835.000	890.000
1546	10 :	954.000	1000.000	992.000	990.000	964.000
1551	15 :	913.000	873.000	812.000	744.000	672.000
1556	20 :	606.000	533.000	472.000	412.000	362.000
1561	25 :	319.000	270.000	236.000	206.000	172.000
1566	30 :	143.000	122.000	103.000	88.000	73.000
1571	35 :	64.000	54.000	47.000	39.000	34.000

MOMENTUM APPROXIMATION

MK( 0)= .170908E+05

MK( 1)= .261753E+06

MD( 0)= .203427E+05

MD( 1)= .397631E+06

ASSESSED DECAY TIME= .423127E+01



DECAY TIME ITERATION

TAU( 1)= .846255E+01  
AMPL. FACTOR= .170546E+00  
AV.DEV.AT 0 TH ITER.= .121387E+03  
SUM BEFORE=-.953603E+03 SUM AFTER= .313807E+04 ALL= .218447E+04

TAU( 2)= .423127E+01  
AMPL. FACTOR= .274366E+00  
AV.DEV.AT 0 TH ITER.= .399036E+02  
SUM BEFORE= .226874E+03 SUM AFTER=-.104410E+04 ALL=-.817222E+03

TAU( 3)= .211564E+01  
AMPL. FACTOR= .520247E+00  
AV.DEV.AT 0 TH ITER.= .144996E+03  
SUM BEFORE= .187289E+04 SUM AFTER=-.317205E+04 ALL=-.129916E+04

TAU( 4)= .557035E+01  
AMPL. FACTOR= .221958E+00  
AV.DEV.AT 0 TH ITER.= .209338E+02  
SUM BEFORE=-.311137E+03 SUM AFTER= .378239E+03 ALL= .671017E+02  
ORDERED SET!

TAU( 1)= .846255E+01 SD= .121387E+03  
TAU( 2)= .557035E+01 SD= .209338E+02  
TAU( 3)= .423127E+01 SD= .399036E+02

TAU( 5)= .527810E+01  
AMPL. FACTOR= .230827E+00  
AV.DEV.AT 1 TH ITER.= .124256E+02  
SUM BEFORE=-.212868E+03 SUM AFTER= .722048E+02 ALL=-.140663E+03  
ORDERED SET!

TAU( 1)= .557035E+01 SD= .209338E+02  
TAU( 2)= .527810E+01 SD= .124256E+02  
TAU( 3)= .423127E+01 SD= .399036E+02

TAU( 6)= .510958E+01  
AMPL. FACTOR= .236471E+00  
AV.DEV.AT 1 TH ITER.= .108263E+02  
SUM BEFORE=-.151978E+03 SUM AFTER=-.105788E+03 ALL=-.257765E+03  
ORDERED SET!

TAU( 1)= .527810E+01 SD= .124256E+02  
TAU( 2)= .510958E+01 SD= .108263E+02  
TAU( 3)= .423127E+01 SD= .399036E+02



TAU( 7)= .502953E+01  
AMPL. FACTOR= .239304E+00  
AV.DEV.AT 1 TH ITER.= .115488E+02  
SUM BEFORE=-.121883E+03 SUM AFTER=-.190668E+03 ALL=-.312551E+03  
ORDERED SET!  
TAU( 1)= .527810E+01 SD= .124256E+02  
TAU( 2)= .510958E+01 SD= .108263E+02  
TAU( 3)= .502953E+01 SD= .115488E+02

TAU( 8)= .514927E+01  
AMPL. FACTOR= .235104E+00  
AV.DEV.AT 1 TH ITER.= .108331E+02  
SUM BEFORE=-.166615E+03 SUM AFTER=-.637790E+02 ALL=-.230394E+03  
ORDERED SET!  
TAU( 1)= .514927E+01 SD= .108331E+02  
TAU( 2)= .510958E+01 SD= .108263E+02  
TAU( 3)= .502953E+01 SD= .115488E+02

TAU( 9)= .509131E+01  
AMPL. FACTOR= .237109E+00  
AV.DEV.AT 1 TH ITER.= .109064E+02  
SUM BEFORE=-.145178E+03 SUM AFTER=-.125137E+03 ALL=-.270315E+03  
ORDERED SET!  
TAU( 1)= .514927E+01 SD= .108331E+02  
TAU( 2)= .510958E+01 SD= .108263E+02  
TAU( 3)= .509131E+01 SD= .109064E+02

TAU( 10)= .512038E+01  
AMPL. FACTOR= .236096E+00  
AV.DEV.AT 1 TH ITER.= .108036E+02  
SUM BEFORE=-.155983E+03 SUM AFTER=-.943404E+02 ALL=-.250323E+03  
ORDERED SET!  
TAU( 1)= .514927E+01 SD= .108331E+02  
TAU( 2)= .512038E+01 SD= .108036E+02  
TAU( 3)= .510958E+01 SD= .108263E+02

TAU( 11)= .512941E+01  
AMPL. FACTOR= .235784E+00  
AV.DEV.AT 1 TH ITER.= .107986E+02  
SUM BEFORE=-.159317E+03 SUM AFTER=-.847779E+02 ALL=-.244095E+03  
ORDERED SET!  
TAU( 1)= .514927E+01 SD= .108331E+02  
TAU( 2)= .512941E+01 SD= .107986E+02  
TAU( 3)= .512038E+01 SD= .108036E+02



TAU( 12)= .513481E+01  
 AMPL. FACTOR= .235599E+00  
 AV.DEV.A1 1 TH ITER.= .108019E+02  
 SUM BEFORE=-.161304E+03 SUM AFTER=-.790742E+02 ALL=-.240378E+03  
 ORDERED SET!  
 TAU( 1)= .513481E+01 SD= .108019E+02  
 TAU( 2)= .512941E+01 SD= .107986E+02  
 TAU( 3)= .512038E+01 SD= .108036E+02

TAU( 13)= .512760E+01  
 AMPL. FACTOR= .235847E+00  
 AV.DEV.A1 1 TH ITER.= .107986E+02  
 SUM BEFORE=-.158645E+03 SUM AFTER=-.867059E+02 ALL=-.245351E+03  
 ORDERED SET!  
 TAU( 1)= .512941E+01 SD= .107986E+02  
 TAU( 2)= .512760E+01 SD= .107986E+02  
 TAU( 3)= .512038E+01 SD= .108036E+02

CALCD DECAY TIME= .512760E+01(+/-) .451469E-02

TIME	K(T)	D(T)	C(T)	D(T)-C(T)	REL.SD
0	0	0	0.000	0.00000	0.00000
1	15	0	1.769	1.76885	0.16380
2	41	4	7.746	3.74575	0.34688
3	128	24	25.446	1.44575	0.13388
4	281	65	66.493	1.49344	0.13830
5	393	138	128.321	11.16950	1.03435
6	595	222	213.882	8.11835	0.75180
7	691	322	315.203	6.79724	0.62946
8	835	438	424.867	13.13280	1.21616
9	890	549	535.559	13.44120	1.24472
10	954	665	639.521	25.47860	2.35945
11	1000	760	736.698	23.30240	2.15792
12	992	840	820.176	19.82410	1.83581
13	990	911	887.851	23.14920	2.14373
14	964	955	940.275	14.72510	1.36362
15	913	980	974.873	5.12695	0.47478
16	873	998	993.676	4.32434	0.40046
17	812	1000	998.072	1.92773	0.17852
18	744	981	987.752	6.75232	0.62530
19	672	963	964.173	1.17249	0.10858
20	606	929	930.001	1.00146	0.09274
21	533	892	886.873	5.12744	0.47483
22	472	849	837.109	11.89100	1.10116
23	412	793	783.169	9.83130	0.91043
24	362	745	727.068	17.93230	1.66062
25	319	682	670.985	11.01510	1.02006
26	270	628	614.888	13.11160	1.21420
27	236	574	559.968	14.03230	1.29947
28	206	521	507.941	13.05880	1.20931
29	172	464	458.213	5.78723	0.53593
30	143	415	410.577	4.42328	0.40962



31	122	367	366.091	-	0.90924	-	0.08420
32	103	330	325.209	-	4.79114	-	0.44368
33	88	292	287.958	-	4.04187	-	0.37430
34	73	257	254.083	-	2.91653	-	0.27009
35	64	223	223.694	-	0.69406	-	0.06427
36	54	196	196.637	-	0.63669	-	0.05896
37	47	168	172.578	-	4.57773	-	0.42392
38	39	145	151.159	-	6.15900	-	0.57035
39	34	121	132.169	-	11.16950	-	1.03435

## 12. KÖSZÖNETNYILVÁNÍTÁS

A szerzők köszönettel tartoznak Hargitai Csabának a kézirat lektorálása során tett észrevételeiért és az előkészítés során folytatott értékes diszkusszióért, valamint Hetzer Emilnének a kézirat rendezésében nyújtott segítségéért.

## 13. IRODALOM

- [1] C.C. Goodyear: Signals and Information Butterworth, London, 1971, 210 old.
- [2] L. Hundley, T. Coburn, E. Garwin, L. Stryer: Rev. Sci. Instr. 1967, 38, 488
- [3] J.N. Demas, G.A. Crosby: Anal. Chem. 1970, 42, 1010
- [4] L.J. Cline-Love, L.A. Shaver: Anal. Chem. 1979, 51, 1941
- [5] J. Brody: Rev. Sci. Instr. 1975, 28, 1022
- [6] A.E.W. Knight, B.K. Selinger: Spectrochim. A. 1971, 27, 1223
- [7] Z. Bay: Phys. Rev. 1950, 77, 419;  
Z. Bay, U.P. Henry, H. Kanner: Phys. Rev. 1955, 100, 1197
- [8] J. Mullooly: Biophys. J. 1973, 13, 1109
- [9] R. Schuyler and I. Isenberg: Rev. Sci. Instr. 1971, 42, 813
- [10] I. Isenberg: J. Chem. Phys. 1973, 59, 5708  
I. Isenberg, R.D. Dyson, R. Hanson: Biophys. J. 1973, 13, 1090
- [11] A. Grinvald, I.Z. Steinberg: Anal. Biochem. 1974, 59, 583
- [12] Szőke J., Láng E.: KFKI-Report 1979-12







- 12/ Szőke J.: Mérési eredmények legkisebb négyzet fittelése  
polinomokkal  
KFKI Report 1977-3
- 13/ Szőke J.: Ocenyka sz pomoscsju EVM krivih zatuhanyija fotonnoj  
emisszii  
KFKI-Report 1978-78
- 14/ Szőke J., Mészáros Gy., Hargittai Cs.: Függvények közelítő  
értékeinek kiszámítása ortogonális polinomokkal  
KFKI-Report 1979-13
- 15/ Szőke J., Láng E.: Véges szélességű impulzusokkal gerjesztett  
lecsengési görbék szimulációja  
KFKI-Report 1979-12
- 16/ Láng E., Szőke J.: A lecsengési görbék kiértékelése során  
szerzett tapasztalatok  
2. Magyar Lumineszcencia Nyári Iskola, Debrecen, 1979  
Előadás kivonatok
- 17/ Szőke J., Láng E.: Komputeres adatfeldolgozás 17. Teljesen  
torzított egykomponensű exponenciális lecsengési görbék ana-  
lizise iterációs eljárással  
KFKI Report 1980-44





Kiadja a Központi Fizikai Kutató Intézet  
Felelős kiadó: Krén Emil  
Szakmai lektor: Hargitai Csaba  
Nyelvi lektor: Hargitai Csaba  
Példányszám: 515 Törzsszám: 80-398  
Készült a KFKI sokszorosító üzemében  
Budapest, 1980. július hó